Web Relation Extraction with Distant Supervision

The University Of Sheffield.

Isabelle Augenstein

The University of Sheffield

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

July 2016

Acknowledgements

First and foremost, I would like to thank my PhD supervisors Fabio Ciravegna and Diana Maynard for their support and guidance throughout my PhD and their valuable and timely feedback on this thesis. I would also like to thank the Department of Computer Science, The University of Sheffield for funding this research in the form of a doctoral studentship. My thanks further goes to my thesis examiners Rob Gaizauskas and Ted Briscoe for their constructive criticism.

During my thesis I had a number of additional mentors and collaborators to whom I am indebted. I am grateful to Andreas Vlachos for introducing me to the concept of imitation learning and his continued enthusiasm and feedback on the topic. My gratitude goes to Kalina Bontcheva, for offering me a Research Associate position while I was still working on my PhD and supporting my research plans. I am thankful to Leon Derczynski for sharing his expertise on named entity recognition and experiment design. I further owe my thanks to Anna Lisa Gentile, Ziqi Zhang and Eva Blomqvist for collaborations during the earlier stages of my PhD. My thanks goes to advisors on my thesis panel Guy Brown, Lucia Specia and Stuart Wrigley, who challenged my research ideas and helped me form a better research plan. I also want to thank those who encouraged me to persue a research degree before I moved to Sheffield, specifically Sebastian Rudolph and Sebastian Padó.

My PhD years would not have been the same without colleagues, fellow PhD students and friends in the department. Special thanks go to Rosanna Milner and Suvodeep Mazumdar for cheering me up when things seemed daunting, and to Johann Petrak and Roland Roller for co-representing the German-speaking minority in the department and not letting me forget home.

Last but not least, my deep gratitude goes to my friends and family and my partner Barry for their love, support and encouragement throughout this not always easy process.

Abstract

Being able to find relevant information about prominent entities quickly is the main reason to use a search engine. However, with large quantities of information on the World Wide Web, real time search over billions of Web pages can waste resources and the end user's time. One of the solutions to this is to store the answer to frequently asked general knowledge queries, such as the albums released by a musical artist, in a more accessible format, a knowledge base. Knowledge bases can be created and maintained automatically by using information extraction methods, particularly methods to extract relations between proper names (named entities). A group of approaches for this that has become popular in recent years are distantly supervised approaches as they allow to train relation extractors without text-bound annotation, using instead known relations from a knowledge base to heuristically align them with a large textual corpus from an appropriate domain. This thesis focuses on researching distant supervision for the Web domain. A new setting for creating training and testing data for distant supervision from the Web with entity-specific search queries is introduced and the resulting corpus is published. Methods to recognise noisy training examples as well as methods to combine extractions based on statistics derived from the background knowledge base are researched. Using co-reference resolution methods to extract relations from sentences which do not contain a direct mention of the subject of the relation is also investigated. One bottleneck for distant supervision for Web data is identified to be named entity recognition and classification (NERC), since relation extraction methods rely on it for identifying relation arguments. Typically, existing pre-trained tools are used, which fail in diverse genres with non-standard language, such as the Web genre. The thesis explores what can cause NERC methods to fail in diverse genres and quantifies different reasons for NERC failure. Finally, a novel method for NERC for relation extraction is proposed based on the idea of jointly training the named entity classifier and the relation extractor with imitation learning to reduce the reliance on external NERC tools. This thesis improves the state of the art in distant supervision for knowledge base population, and sheds light on and proposes solutions for issues arising for information extraction for not traditionally studied domains.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

In the information age, we are facing an abundance of information on the World Wide Web through different channels – news websites, blogs, social media, just to name a few. One way of making sense of this information is to use search engines, which locate information for a specific user query and sort Web pages by relevance. This still leaves the user to dig through several Web pages and make sense of overlapping, and sometimes even contradictory pieces of information.

These problems have partly been addressed by *information extraction (IE)*, an area which aims at capturing central concepts in text, such as proper names or relations between them, and research in the area of *knowledge base construction and population*, which aims at modelling and storing such world knowledge.

Both knowledge bases and information extraction methods can help to answer user queries more effectively. For instance, if a user query is to obtain the names of all albums by "The Beatles", this information could already be stored in the knowledge base and then retrieved from there, or information extraction methods could extract this information from the Web. Table 1.1 shows a portion of the information contained in the knowledge base Freebase (Bollacker et al., 2008) about The Beatles. The header indicates that The Beatles have a unique identifier (mid), one or several types (here, Musical Artist) and several relations such as genres, place musical career began, albums or record labels.

Because it saves time and resources to retrieve common facts from knowledge bases, they have become a popular solution for Web search, e.g. Google uses the Google Knowledge Vault (Dong et al., 2014) to enhance search. Instead of manually populating knowledge bases, which can be very laborious and expensive, IE methods can then be used for *automatic knowledge base population (KBP)*. Entities in knowledge bases can further be used for entity disambiguation and identification in unstructured text (Bunescu and Pasça, 2006; Mendes et al., 2011; Zheng et al., 2012). Relations in knowledge bases enable more complicated natural language processing tasks such as question answering (Yao and Van Durme, 2014; Fader et al., 2014), also used in industry settings such as the IBM DeepQA question answering framework (Wang et al., 2012a).

The Beatles, mid: /m/07c0j, notable type: Musical Artist	
Relation	Relation Value
Musical Genres	Rock music, Pop music, Pop rock, Psychedelic rock
Place Musical Career Began	Liverpool
Albums	Red Album, Something New The Beatles, Please Please Me, Let It Be
Record Labels	Capitol Records, Parlophone, Apple Records, EMI, MGM Records

Table 1.1: Information about The Beatles in the knowledge base Freebase

Early work in information extraction has focused on researching methods recognising proper names, i.e. *named entities (NEs)* such as names of politicians, for competitions including MUC (Grishman and Sundheim, 1995) or ACE (Doddington et al., 2004), which provided hand-labelled training data and fixed schemas to define what type of information to extract, similar to schemas in knowledge bases such as Freebase. As a result, many of those early information extraction methods were *supervised* and required manually labelled data. With the emergence of the Web and more heterogeneous domains and types of text, *unsupervised* methods became popular, which do not require a fixed schema and thus do not make any strong assumption about the content of text (e.g. Etzioni et al. (2004)). Instead, they provide methods to group ("cluster") similar information and to explore the resulting clusters. While unsupervised methods are useful for exploring information, they are not as useful for knowledge base population, where the goal is to extract information with respect to a schema.

A further approach for relation extraction called *distant supervision* combines the benefits of the two streams of information extraction approaches (Craven et al., 1999; Mintz et al., 2009). It is a weakly supervised *relation extraction (RE)* method which allows one to extract relations with respect to an existing schema, but does not require manually annotated training data. Instead, it requires a partly populated knowledge base already containing some examples for each relation, and a large corpus of the same domain. The approach both achieves state of the art results in recent evaluation campaigns (Surdeanu and Ji, 2014) and has been shown to be useful in an end-to-end real world setting (Dong et al., 2014).

The approach relies on a heuristic labelling method to automatically generate training data for a supervised classifier: it finds sentences which contain two named entities which, according to the knowledge base, are in a relation, and assumes they are positive training data for that relation. If a sentence contains two NEs which are not in a relation according to the knowledge base, the sentence is used as negative training data. Assuming a partly populated knowledge base with the NE pair "The Beatles", "Capitol Records", which is an example for the relation *Musical Artist: record label*, Example 1.1[1] illustrates how a sentence can be annotated using the distant supervision assumption. Example 1.2 further shows an ambiguous example, as explained later in this section, for which the distant supervision heuristic fails, here annotated correctly with the

[1] http://h2g2.com/approved_entry/A3418201

relation *Musical Artist: album* between the NEs "The Beatles" and "Let It Be".

(1.1) In November 1963 *Capitol Records* finally signed a contract with *the Beatles* and announced plans to release the Beatles' single 'I Want To Hold Your Hand' in December 1963 as well as their second album With The Beatles in January.

(1.2) *Let It Be* is the twelfth and final album by *The Beatles* which contains their hit single 'Let it Be'. They broke up in 1974.

An important characteristic of a successful knowledge base population method is to discover new facts with high precision, e.g. 0.9 or above (Dong et al., 2014). State of the art relation extraction methods from unstructured text do not achieve this performance, e.g. on the KBP 2014 slot filling challenge, the best system achieved a precision of 0.5540 at a recall of 0.2814 (Surdeanu and Ji, 2014). Currently it is difficult to achieve a successful knowledge base population method as described above; successful approaches such as Dong et al. (2014) rely on combining different extraction methods, extract from several sources and adjust the confidence threshold for extraction.

This thesis aims at researching a relation extraction approach from Web text which allows one to extract new facts with high precision and does not require any manually labelled data. To achieve this, several challenges distant supervision approaches face are discussed and addressed:

- One of the main challenges of distant supervision is that the heuristic for automatically annotating training data sometimes fails, which leads to noise. In Example 1.2, the first mention of *Let It Be* is an example for the *MusicalArtist:album* relation, whereas the second mention is an example of the *MusicalArtist:track* relation (see Table 1.1). Using the distant supervision assumption, both of those would be used as a positive example for *MusicalArtist:track*, although only one of them is a true positive. Using noisy training data like that results in a lower precision than using manually annotated training data. Thus, using methods for identifying such noise is crucial to improving precision of distant supervision approaches.

- A further restriction of distant supervision methods is that, at extraction time, only sentences which contain two named entities are considered candidates for extraction. This means that both subjects (e.g. "The Beatles") and objects ("Let It Be") of relations need to be referred to by the proper name in order to be a valid extraction candidate. However, authors do not write their articles in such a manner – they might mention the name of the subject in the first sentence and in following sentences refer to it with pronouns or nouns, e.g. "they" or "the band". Ignoring this means potential new facts can be missed (Gabbard et al., 2011).

- When populating a knowledge base, evidence for the same facts may appear several times on different Web pages. This can be utilised to improve the overall precision of extractions. Intuitively, the more often the same information can be observed, the more likely it is to be true, and the more different sources it appears in, the stronger the evidence is. Another challenge is therefore how to make use of this in order to improve distant supervision performance.

- A prerequisite for relation extraction is to recognise named entities which are potentially arguments of relations. Named Entity Recognition and Classification (NERC) for newswire text has been studied extensively, e.g. in the context of challenges such as MUC (Grishman and Sundheim, 1995) or ConLL (Tjong Kim Sang and De Meulder, 2003) and off-the-shelf NERC tools tuned for specific training corpora are available as a result (Finkel et al., 2005). However, large hand-labelled NERC corpora are mostly available for the newswire genre and existing tools struggle to generalise over text from genres they were not trained on.

- A corpus of Web pages for relation extraction has the potential to be very diverse, spanning different domains and containing articles written by different authors. When named entity recognisers fail on such text, it would be useful to understand what the reason for that failure is. For example, is the main reason that named entities are not seen in the training set or that the context is very diverse? Such studies could help to inform how to better develop NERC methods for relation extraction from Web pages.

- Assuming existing NERC approaches fail for the Web genre, what could an alternative approach be? Ideally, a relation extraction approach should not be completely dependent on the performance of external tools such as 3rd party named entity recognisers and classifiers. Further, it is desirable to have a NERC approach which is robust across domains and genres does not require any additional manual annotation.

- Finally, in order to achieve high precision for distant supervision approaches from Web pages, it is important to have expressive features which can capture indicators for relations. Some features may be high-frequency, but low precision, i.e. they may appear in the context of specific relations very often, but may also appear often in other contexts. Other features can be the opposite, they do not appear often, but when they do, they are very good indicators for certain relations. A challenge is therefore to study and select the best combination of such features. Web pages also contain more than just unstructured text – they often contain markup such as hyperlinks and lists. The usefulness of those characteristics for relation extraction is studied in addition.

1.2 Contribution

This thesis presents novel methods to address the challenges described in the previous section. In more detail, the contributions of this thesis are as follows:

- Statistical methods for reducing noise in automatically generated training data for distant supervision, based on assessing the ambiguity of relation examples. Results show that those methods outperform a baseline method without filtered training data.

- Evaluating methods for extracting relations from sentences which do not contain a direct mention of the subject of the relation, and thus potentially discovering more new facts. Methods are based on existing co-reference resolution methods and additional co-reference resolution heuristics. Results indicate that, although extractions obtained via co-reference

resolution may be more prone to noise, they increase precision overall by providing more additional results which can be combined for knowledge base population.

- Evaluating methods for integrating extracted relations based on either integrating features of mentions across sentences and documents or integrating extractions post-hoc. Results show that integrating extractions post-hoc leads to a higher precision. Further, using statistics gathered from the knowledge base about the number of results per relation and subject entity and about which relations often have objects with the same surface form improves precision.

- Developing an entity-centric search-based approach for Web relation extraction with distant supervision which utilises a search engine to gather training data for relation examples given the subject of the relation and the relation name. The approach has the benefit that it is easy to find positive training examples in the resulting Web pages and negative training data can be sampled from the same Web pages.

- Studying reasons for poor named entity recognition and classification performance in different genres. A quantitative analysis finds that one of the main reasons for low NERC performance is NEs which appear in the testing, but not the training set. Another reason is that NERC corpora for diverse genres are small and, because popular NEs change over time, it is difficult to maintain them. This impacts NER performance and in turn relation extraction performance, especially if off-the-shelf NER tools, which are tuned to perform well on newswire corpora, are used as a pre-processing step for relation extraction

- Proposing a joint approach for named entity recognition and classification and relation extraction based on imitation learning, a structured prediction method. This significantly outperforms distant supervision approaches with two off-the-shelf supervised NEC systems, Stanford NER and FIGER, one of which is trained on newswire, and the other one of which is trained on a very similar genre, Wikipedia.

- Evaluating and comparing different methods against a distant supervision approach with imitation learning: distant supervision with off-the-shelf supervised NEC, as mentioned above, a relation exraction approach without NEC preprocessing, and a one-stage classification approach which aggregates named entity and relation features.

- Exploring and evaluating the effect of different named entity classification and relation extraction features, including Web features. Low-frequency high-precision features such as parsing features lead to higher average precision than high-frequency features such as the bag of context words. Web-based features, e.g. occurrence in a list or hyperlinks significantly improve average precision.

To summarise, the contributions of this thesis are to study methods for selecting training and testing data for Web-based distant supervision; to research reasons for NERC failure in diverse genres; and to propose a method for jointly learning a NERC and a RE which do not rely on manually labelled data.

1.3 Thesis Structure

The remainder of this thesis is structured as follows:

- **Chapter 2** describes the task of relation extraction and the notion of knowledge bases. Different streams of research on relation extraction for knowledge base populatio without manually annotated data are described and compared and the choice of distant supervision as a research method is motivated. Existing work on distant supervision and its limitations are discussed in depth and the research presented in this thesis is motivated based on this analysis of the state of the art.
- **Chapter 3** describes the aims of the research conducted within the context of this thesis, as well as how the different contributions fit together to advance the state of the art in information extraction.
- **Chapter 4** describes experiments for reducing noise of heuristically labelled training data, for extracting relations across sentence boundaries with co-reference resolution, and methods for relation integration. It introduces the entity-centric search-based approach for distant supervision, which is also used for subsequent work. The chapter shows early results for how NER for distant supervision can be improved by using part-of-speech-based and Web-based heuristics in addition to preprocessing with Stanford NER and concludes that NERC is one of the bottlenecks for Web-based distant supervision.
- **Chapter 5** describes a qualitative study, analysing reasons for failure of different existing NERC approaches in diverse genres. The chapter concludes with lessons learnt for NERC for Web-based distant supervision.
- **Chapter 6** then proposes a solution for the problem of NERC for distant supervision. A novel method or distant supervision with imitation learning, which jointly learns models for NEC and RE. The approach is compared to distant supervision with supervised off-the-shelf NER approaches, as studied in Chapter 4. It further documents research on features for named entity and relation extraction, including Web features.
- **Chapter 7** summarises the work of this thesis and suggests future work directions.

1.4 Previously Published Material

The major research documented in this thesis has been published in conference proceedings and journals or is currently under review as follows:

- Parts of **Chapter 3** have been published in the proceedings of the 13th International Semantic Web Conference (Augenstein, 2014a).
- **Chapter 4** is based on publications at the third workshop on Semantic Web and Information Extraction at the 25th International Conference on Computational Linguistics (Augenstein, 2014b), in the proceedings of the 19th International Conference on Knowledge Engineering

and Knowledge Management (Augenstein et al., 2014), and in the Semantic Web Journal (Augenstein et al., 2016a).

- **Chapter 5** is based on work which is currently under review with the journal Information Processing & Management (Augenstein et al., 2015a).
- **Chapter 6** is based on a publication in the proceedings of the 20th Conference on Empirical Methods in Natural Language Processing (Augenstein et al., 2015b).

For the publications listed above, the first author proposed and conducted the experiments and was supported by the co-authors with discussions and guidance on the work as well as with feedback on the writing of the publications.

Other papers which have been published in the timeframe of this thesis and have served as motivation for some of the work presented in this thesis, though do not have any significant overlap with chapters in this thesis, are: Augenstein et al. (2013); Gentile et al. (2013); Zhang et al. (2013a,b); Blomqvist et al. (2013); Zhang et al. (2014, 2015); Derczynski et al. (2015a); Lendvai et al. (2016a,b); Augenstein et al. (2016c,b).

Chapter 2

Background on Relation Extraction

2.1 Introduction

Information extraction is the process of extracting information and turning it into structured data, or also the activity of populating a structured knowledge source with information from an unstructured knowledge source (Gaizauskas and Wilks, 1998). The information contained in the structured knowledge base can then be used as a resource for other tasks, such as answering natural language queries or text mining. There are many different kinds of information which can be extracted, for example proper names and relations between them.

This chapter contains a description of the background and related work for relation extraction without manually annotated training data. First, an overview of the relation extraction task is given in Section 2.2, which defines relation extraction formally and describes a typical relation extraction pipeline. In Section 2.3, relevant research streams which fit the scope of the thesis, i.e. relation extraction methods for knowledge base population which do not require manual annotation, are described and the choice of research stream for the remainder of the thesis, distant supervision, is motivated. This is followed by a detailed survey of research on distant supervision with respect to 5 different aspects in Section 2.4. The chapter concludes with an analysis of research gaps in the state of the art. Since only relation extraction research streams that fit the scope of this thesis are described, existing work on two big research streams is not discussed, namely rule-based methods and supervised methods. For a survey on those and a comparison to the relation extraction research streams discussed in this thesis, the reader is referred to Bach and Badaskar (2007).

2.2 The Task of Relation Extraction

2.2.1 Formal Definition

Relation extraction (RE) is defined as the task of extracting semantic relations between arguments (Bach and Badaskar, 2007). Arguments can either be general concepts such as "a company" (ORG), "a person" (PER); or instances of such concepts (e.g. "Microsoft", "Bill Gates"), which are called proper names or named entitites (NEs). An example for a semantic relation would be "PER founder-of ORG", also written as "founder-of(PER, ORG)". Semantic relations further contain predicates (e.g. "founder-of"), also sometimes called "properties". Note that there is some degree of confusion in the field regarding the terms "relation", "property' and "predicate'. In this thesis, the terms "predicate" and "property" is used to refer to the name of the relation (e.g. "founder-of"), whereas the term "relation" will be used to refer to <subject, predicate, object> tuples. Conceptual knowledge as well as instance definitions and relations are stored in a knowledge base.

Formally, let concepts (also called classes) be defined as C, and instances of such classes, also called entities, as E. In this thesis, relations are extracted with respect to a knowledge base KB such as Freebase (Yao and Van Durme, 2014), in which each concept $c \in C$, each entity $e \in E$ and each property $p \in P$ is considered a resource $r \in R$ with a unique identifier. Resources are a way of assigning a unique identifier to all things described in a knowledge base.

Concepts are named entity types; traditional coarse-grained types used are person (PER), organisation (ORG), location (LOC), date (DATE) and miscellaneous (MISC). In addition, each resource $r \in R$ has a set of lexicalisations, $L_r \subset L$. Lexicalisations in the KB are typically represented as either a name or an alias, i.e. a less frequent name of a resource. As an example, the entity with unique Freebase identifier `http://www.freebase.com/m/017nt` has the name "Bill Gates", aliases including "William H. Gates III", and the coarse type `/people/person`. More details on knowledge bases are given in Section 2.2.3.

Further, although relations can have more than two arguments, only binary relations or relations which can be expressed in binary form are considered here. For instance, "has siblings" can either be expressed as an n-ary relation with n+1 being the number of siblings, or as several binary relations (Surdeanu and Ji, 2014). In the TAC KBP 2014 Slot Filling challenge, these are called "list" relations, and are expressed as several binary relations. On the other hand, some relations are 3-ary or 4-ary because they contain a date and/or place as additional arguments, e.g. "PER married PER on DATE at LOC". In that case the relation could not simply be expressed as several binary relations and all arguments past the second one would be disregarded for the purpose of relation extraction.

Binary relations consist of a subject (e.g. "Bill Gates"), a predicate (e.g. "founder-of") and an object (e.g. "Microsoft"), i.e. they are represented as triples of the form $(s, p, o) \in E \times P \times E$.

2.2.2 Relation Extraction Pipeline

This subsection aims at describing a typical relation extraction approach. A graphical overview of such an RE pipeline is given in Figure 2.1. Note that there are several variations of this approach,

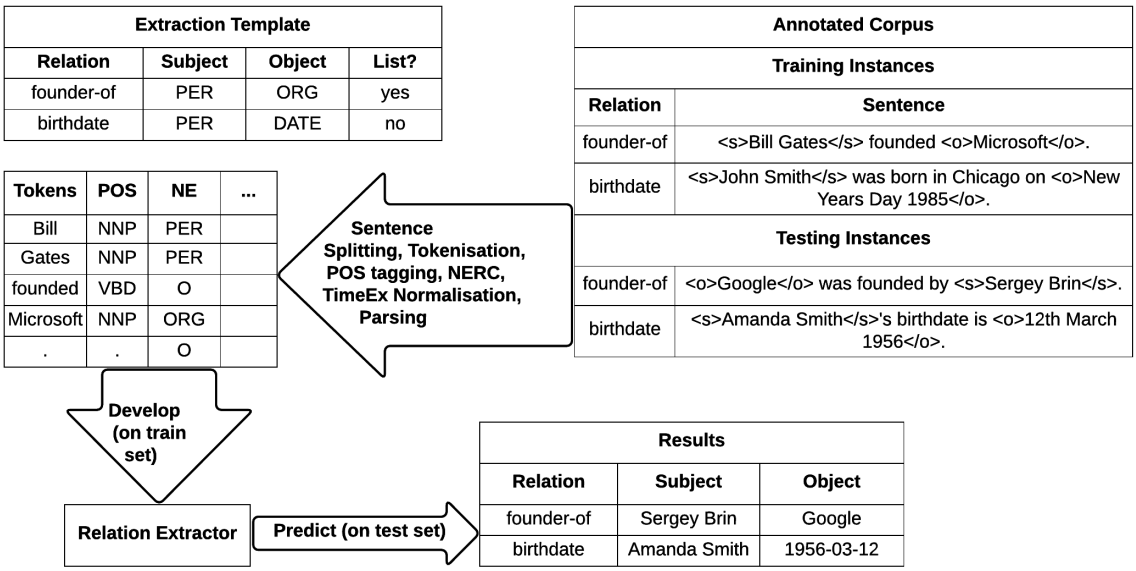

Figure 2.1: Typical Relation Extraction Pipeline

as described in the following chapters.

The input to the relation extraction task is usually a set of training documents, a set of testing documents and an extraction template. The extraction template defines which relations are to be extracted and how they are defined, i.e. how many arguments they have and what concepts those arguments belong to. For instance, "founder-of" is defined as a relation between a person (PER) and an organisation (ORG): founder-of(PER, ORG) and is a "list" relation, i.e. may have more than one object (founder) per subject and relation. Detailed NE types are not always given, e.g. the TAC KBP 2014 Slot Filling challenge does not provide the NE type of the object of the relation (Surdeanu and Ji, 2014). Next, the documents are pre-processed with several NLP steps to determine morphology, syntax and semantics of the sentence. These pre-processing steps aim at helping to "understand" text to facilitate the extraction of relations. One of the most important pre-processing steps is named entity recognition and classification (NERC), which is the task of recognising and assigning a type to proper names in text. This is because, as already mentioned in the previous section, relations are either extracted between named entities only or between a mixture of named entities and general concepts ("a person"). As an example, "Bill Gates" would be assigned the type "PER" and "Microsoft" the type "ORG". Historically, the first series of NER evaluation efforts at the MUC conferences distinguished between the named entity types person (PER), location (LOC), organisation (ORG) and miscellaneous (MISC) (Grishman and Sundheim, 1995), though depending on the extraction template, more fine-grained types (e.g. Politician, Film) may be used.

After pre-processing, the training set is used to develop relation extractors, after which they are applied to the test set to extract relations. If more than one relation per relation template is extracted, those extractions are validated. The definition of relations can help with this. For

instance, a company may have more than one founder, but every person only has two biological parents, which determines how many extractions per subject of each relation should be returned. The output of the relation extraction task is a set of annotated test documents (often called *sentence-level extraction*) or a list of extraction triples (*instance-level extraction*). In case the output is a list of extractions, those can be used to populate knowledge bases. More details on knowledge bases and their role in the relation extraction task are explained in the next section.

2.2.3 The Role of Knowledge Bases in Relation Extraction

Knowledge bases are an integral part of the relation extraction process. They consist of a schema, sometimes also called extraction template, and data associated with the schema. The schema defines the structure of information, e.g. it might define that persons can be politicians or musicians, and that they have names and birthdates, politicians are in addition associated with a party and musicians play instruments in bands with other musicians. In other words, the schema defines *classes* (e.g. Person), their *subclasses* (e.g. Politician) and *properties* (e.g. in-party). What is relevant for the task of relation extraction is that properties define what relations can hold between instances of classes, whereas their classes restrict the types of the relations' arguments. The data associated with the schema would then be examples of such politicians and musicians with their respective names, birthdates, parties, instruments and bands. The relation extraction process typically starts with such a schema and the goal is then to annotate text with relations, or to populate the knowledge base with information, i.e. extract and add data. The latter is called *knowledge base population (KBP)* and has become popular, among other reasons, due to the TAC KBP series of challenges[1]. This series of evaluation efforts comprises several parts of the relation extraction pipeline, including extracting relations (slot filling) (Surdeanu and Ji, 2014) and validating extractions (slot filler validation). For slot filling, the subjects of relations are already given and the task is then to find the objects of relations in a corpus.
Shared task evaluation efforts often use locally defined templates. However, with the rise of the World Wide Web and then the Semantic Web, Web-based publicly available knowledge bases also became popular for the KBP task (Ji and Grishman, 2011; Mintz et al., 2009). Since this thesis focuses on such knowledge bases, this subsection contains background on the Semantic Web and Linked Data, the idea of interlinking data in different knowledge bases.

The Semantic Web

The *Web of Data* or *Semantic Web* is a global information space consisting of billions of interlinked documents. Certain Semantic Web standards have been developed by the *World Wide Web Consortium* (W3C)[2] which include *RDF* and *OWL* to describe data and *SPARQL* to query data. These standards can be used to describe the relationships between things, such as people, locations or organisations.

[1] http://www.nist.gov/tac/2014/
[2] http://www.w3.org/standards/semanticweb/

A set of best practices for publishing data on the Web (also called the *Linked Data* principles) were outlined as follows by Tim Berners-Lee [3]:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
4. Include links to other URIs so that they can discover more things.

There are two kinds of information which can be described. First, information about classes (e.g. artist, track) and their relationships (e.g. released-track). This kind of information is published as a *schema*. Second, information about instances of these classes (e.g. David Bowie, Changes) can be published in a *dataset*. Note though that this is optional: some websites contain Semantic Web annotations, but do not publish them in a separate dataset.

While, for the purpose of relation extraction, schemas serve a similar purpose as locally defined templates (Section 2.2.3), there is a clear advantage in the way data is described (using URIs). Imagine a slot filling task, for which the subjects of relations are given and the goal is to extract values for the objects of those relations. Some of the subjects may be ambiguous and refer to many different real-world entities. This ambiguity may be across classes (a jaguar can be an animal or a car brand), or instances of classes may be ambiguous (there are many persons named John Smith). Especially for the latter, it is very useful to have URIs as input for each subject. For instance, if the task is to extract birthdates, the RE approach would be expected to return only one result per subject entity, but would likely find more than one for "John Smith". If there are several URIs with the name "John Smith" in the knowledge base, the RE approach can make use of this information and return several results, or, if other information about persons named "John Smith" is already contained in the knowledge base, try to return the likely birthdate for that specific John Smith, based on that additional information.

More advantages of Semantic Web standards are clarified after taking a closer look at Linked datasets.

Linked Datasets

Since the vision of the Semantic Web was introduced, billions of triples in hundreds of interlinked datasets, describing instances of classes and relations between those instances, have been created. Some of those datasets are released publicly and are available to everyone, in which case they are referred to as *Linked Open Data* (LOD). To get a better idea of the nature and size of Linked Open Data, a visualisation of datasets and their links is shown in Figure 2.2.

The figure shows that there are several cross-domain datasets, with DBpedia having the most links to other datasets and effectively functioning as a hub for Linked Data. Other prominent examples of cross-domain datasets include Freebase (Bollacker et al., 2008), Yago (Suchanek et al., 2008), and Wikidata (Vrandečić and Krötzsch, 2014) (not included in cloud diagram). Domain-specific datasets exist for several different domains: Governments release their data using Semantic

[3] http://www.w3.org/DesignIssues/LinkedData.html
[4] http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/

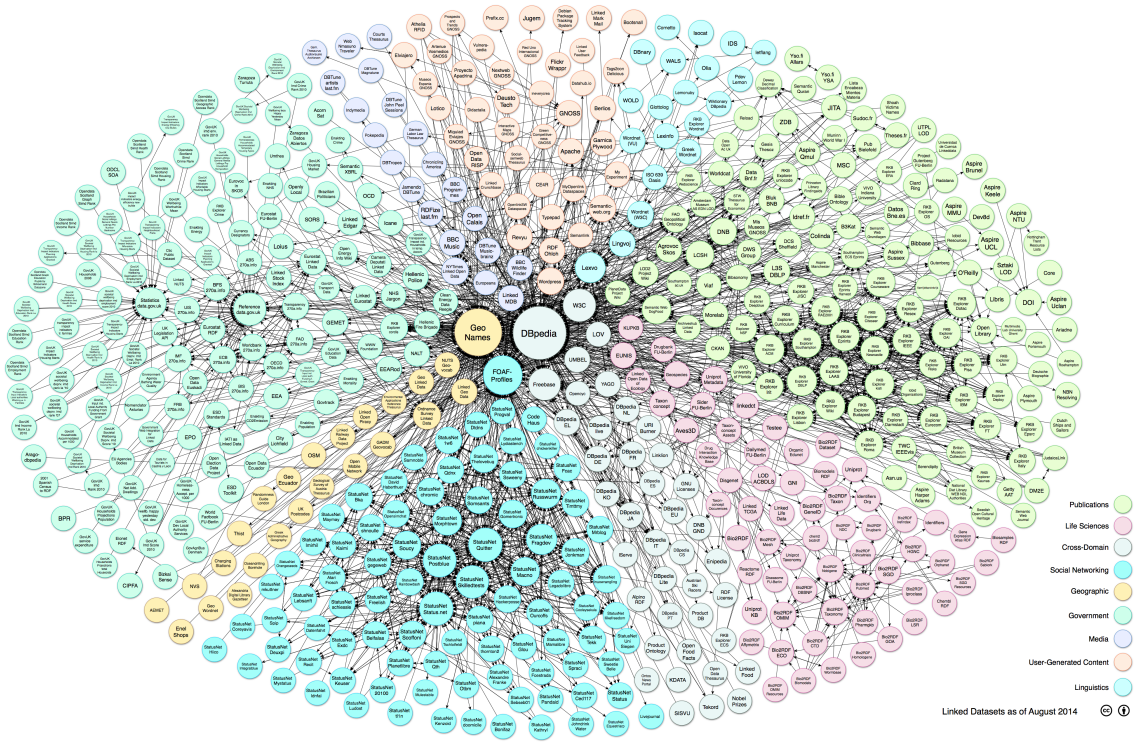

Figure 2.2: LOD Cloud diagram, as of April 2014[4]

Web standards, sciences make use of the technology to describe complex processes with ontologies, libraries and musea structure and release their data about books and artifacts, and media and social media providers enrich their Web sites with semantic information.

There are several strategies for creating Linked datasets: they can be created automatically from existing resources (DBpedia, Yago, WordNet (Fellbaum, 1998)), created collaboratively by a community (Freebase, Wikidata) or they can be created manually by domain experts (MusicBrainz, UMLS). A very popular existing resource is Wikipedia, since it contains both text and also the most important facts for each page summarised in an *infobox* table, which can be converted into a stuctured data format very easily.

One relation extraction method, distant supervision (see Section 2.3.3), relies to a large degree on both the schema and the data contained in Linked datasets. More details on which knowledge bases are used for distant supervision and why are given in Section 2.4.1.

What is important to know for relation extraction is that information in different datasets is often interlinked. Some of them contain information about the same entities and to indicate this, links between datasets exist. This means relation extraction approaches which make use of information already contained in datasets (as becomes clear in Section 2.3 and Section 2.4) can combine information from several datasets. More than that, there are also links on the schema level (e.g. the property "birthdate" in one schema may be linked to the property "born-on" in another schema, or the class "album" may be linked to the class "musicalbum"), which facilitates

combination of information in datasets, and also combination of extraction schemas even further. For instance, one schema may define that musical artists have birthdates, and another that they release albums. Those could then be combined for extracting both relations.

2.3 Relation Extraction with Minimal Supervision

Having seen how a typical relation extraction approach works, this section now details different relation extraction streams which are variations of the typical relation extraction approach described in the previous section. The research streams described here are all the ones which fit the scope of this thesis, i.e. relation extraction methods for knowledge base population from text on Web pages which do not require any, or only very little, manual effort for training. Streams of approaches which fit this scope are *semi-supervised bootstrapping methods*, *unsupervised / Open IE methods* and *distantly supervised* approaches[5].

2.3.1 Semi-supervised Approaches

Semi-supervised or bootstrapping approaches were among the first machine learning based relation extraction approaches, prominent pioneers being DIPRE (Brin, 1999) and Snowball (Agichtein and Gravano, 2000). A description of DIPRE is now given, since subsequent approaches used a similar architecture.

Algorithm 1 DIPRE (Brin, 1999): extract(R, D)

while $|R| < n$ **do**
  $O \leftarrow \text{findOccurrences}(R, D)$
  $P \leftarrow \text{generatePatterns}(O)$
  $R \leftarrow M_D(P)$
end while
return R

DIPRE consists of four simple steps (see Algorithm 1). The input to DIPRE is R, a set of 5 $< s, o >$ tuples for the relation "PERSON author-of BOOK", and D, a document collection, in this case the Web. The first step is to find occurrences of tuples on the Web. Next, patterns are generated. Third, pattern matches are generated; $M_D(p)$ is the set of tuples for which any of the patterns $p \in P$ is matched on a Web page. This process is repeated until n relation occurrences are found.

This basic algorithm is used by almost all semi-supervised approaches, with slight variations. For instance, the input to the algorithm might be examples as well as extraction patterns or extraction rules. Also, matching of patterns can be handled in different ways, using exact or inexact matching. The most interesting part of the algorithm is how patterns are generated. In DIPRE, this is very basic: a pattern is created by grouping sentences for which the sequence of words between person and book match and for which person and book are in the same order.

[5]These methods are summarised here as "minimal supervision" methods, though there is no one agreed upon name for those approaches in the research community

Next, specificity is measured: if a pattern matches too many sentences and as a result specificity is above a manually tuned threshold t, the pattern is rejected. If specificity is too low and only the same book is found with that pattern, the pattern is rejected too.

This already hints at one downside of bootstrapping approaches called *semantic drift*, which is their tendency to move too far away from R and create patterns which express different, related relations, which often co-occur with the same entity tuples, e.g. for "author-of", this could be "editor-of".

Subsequently, bootstrapping models have been researched to improve on the DIPRE model. Prominent large-scale bootstrapping models include KnowItAll (Etzioni et al., 2004) and NELL (Carlson et al., 2010a).

KnowItAll (Etzioni et al., 2004) is a Web-scale information extraction system, which relies on the scale and redundancy of the Web to provide enough information and validate it. In contrast to DIPRE, it does not start with a single relation, but with several, and also contains methods for extending the extraction schema. KnowItAll consists of four modules: an extractor, a search engine interface, an assessor and a bootstrapping component. The *extractor* component applies Hearst patterns (Hearst, 1992) to extract instances for classes (this would be instances of books in DIPRE). Hearst patterns are lexico-syntactic extraction rules such as "NP1 is a NP2", where NP2 is the name of a class such as "books", and NP1 is the name of an instance of that class. Using the search engine interface, these patterns (with NP1 left blank) are then formulated as search queries to retrieve Web pages containing NP1. The component further contains relation extraction rules, e.g. "NP1 plays for NP2", representing the relation "playsFor(Athlete, SportsTeam)". Once all extraction rules are applied, extracted patterns are validated by the *assessor*. The assessor measures co-occurrence statistics of candidate extractions with *discriminator phrases*, which are highly frequent extraction patterns. This means for each search query, e.g. "Tom Cruise starred in X" the number of search results is recorded and the PMI (pointwise mutual information) of the entity, "Tom Cruise" and the pattern is computed. KnowItAll then uses *bootstrapping* in combination with the assessor to validate extractions: for each class, the 20 instances with the highest average PMI are retrieved. These are then used to train conditional probabilities for each extraction pattern. Negative instances are sampled from positive instances for other classes. The best 5 extraction patterns for each class are saved, the rest are discarded. A Naive Bayes classifier is then trained combining evidence from those 5 extraction patterns to classify if an entity, e.g. "Tom Cruise", is an instance of a class, e.g. "actor". Instead of just selecting the best extraction patterns once, a bootstrapping process can be used: once the best 5 extraction patterns are determined, those can be used to find a new set of instances with high PMI. To ensure high quality extraction patterns, incorrect instances are also removed manually. Etzioni et al. (2004) argue that their approach of using Web-based statistics is very useful to discard unreliable patterns; however, they do not measure specificity, as Brin (1999) does. Overall, however, their approach is much more extensible and relies on Web statistics and machine learning instead of just local statistics and patterns.

NELL (Carlson et al., 2010a) is a bootstrapping system that extracts information from the Web to populate a knowledge base and learns to extract information more accurately over time. Like

KnowItall, it is based on the hypothesis that the large amount of redundant information on the Web is a huge advantage for learning mechanisms. The main differences are that the bootstrapping component is more sophisticated and that NELL combines extractions from different sources on the Web: text, lists and tables. Similar to KnowItAll, it learns to extract which instances belong to which classes and which relations hold between instances of those classes. Information is extracted from unstructured information on the Web (i.e. text), as well as semi-structured data (i.e. lists and tables). Extractors are trained in concert using coupled learning based on CPL for free text and CSEAL for lists and tables (Carlson et al., 2010b). CPL, similarly to KnowItAll, relies on co-occurrence statistics between noun phrases and context patterns to learn extraction patterns. CSEAL uses mutual exclusion relationships to provide negative examples, which are then used to filter overly general lists and tables. In addition, NELL learns morphological regularities of instances and probabilistic Horn clause rules to infer new relations from relations it has already learnt. For learning morphological regularities, NELL uses a *coupled morphological classifier (CMC)*. For each class, a logistic regression model is trained to classify noun phrases based on morphological and syntactic features (e.g. words, capitalisation, affixes, POS tags). The *Rule Learner* learns probablistic Horn clauses to infer new relations from relations that are already present in the knowledge base.
The learning system starts with a knowledge base (123 classes, 55 relations, and a few instances for classes and relation triples) and gradually populates and extends it. After the extraction component has extracted a belief, the precision of the belief is evaluated by consulting external data resources or humans, promoting the most strongly supported beliefs to facts and integrating them into the knowledge base. For the following extraction steps, the extractor always uses the updated knowledge base. Carlson et al. (2010a) find that NELL allows one to extract instances and relation with a relatively high precision initially, and that the different extractors they use are complementary. However, one of their findings demonstrates a problem that is very typical of bootstrapping approaches: extraction precision declines over time. In their case it declines from 0.91 to 0.57 over the course of 66 iterations. They suggest that this could be solved by allowing a human to interact with the system during learning using active learning, which was then researched subsequently (Pedro and Hruschka Jr, 2012).

2.3.2 Unsupervised Approaches

Unsupervised relation extraction approaches became popular soon afterwards with *open information extraction* systems such as TextRunner (Yates et al., 2007), ReVerb (Fader et al., 2011) and OLLIE (Mausam et al., 2012). Open information extraction is a paradigm to use simple and scalable methods to extract information which is not restricted beforehand. This is in contrast to semi-supervised approaches described in the previous section, which use pre-defined extraction schemas. Thus, Open IE can be seen as a subgroup of unsupervised approaches. For Open IE methods this means they have to infer entities, their types, and relations between entities from text. As for bootstrapping method, the first Open IE approach is now described to introduce the research stream, and shortcomings and improvements of subsequent research are pointed out.

TextRunner (Yates et al., 2007) is the first fully implemented and evaluated Open IE system.

It learns a Conditional Random Field (CRF) model for relations, classes and entities from a corpus using a relation-independent extraction model. First, it runs over the whole corpus once and annotates sentences with POS tags and noun-phrase chunks. To determine whether a relation should be extracted or not, the system uses a supervised classifier. This supervised classifier is trained by parsing a small subset of the corpus and then heuristically labelling sentences as positive (trustworthy) and negative (not trustworthy) examples using a small set of hand-written rules. The classifier then makes the decision for unseen sentences based on POS tags instead of the parse tree, because it would be too expensive to parse the whole corpus. To resolve synonyms, TextRunner performs unsupervised clustering of relations and entities based on string and distributional similarity (Yates et al., 2007).

ReVerb (Fader et al., 2011) addresses two shortcomings previous Open IE systems have: incoherent extractions and uninformative extractions. Incoherent extractions occur when the extracted relation phrase has no meaningful interpretation. This is due to the fact that decisions in TextRunner are made sequentially. An example would be the relation "contains omits" which is extracted from the sentence "The guide contains dead links and omits sites." To solve this, syntactic constraints on which relations to extract are introduced. The first is that a relation phrase either has to be a verb (e.g. "invented"), a verb followed by a preposition (e.g. "located in") or a verb followed by nouns, adjectives or adverbs and a preposition (e.g. "has atomic weight of"). Also, if there are multiple possible matches, the longest possible match is chosen. If adjacent sequences are found (e.g. "wants", "to extend"), these are merged (e.g. "wants to extend"). Lastly, the relation has to appear between the two arguments in a sentence.
Uninformative extractions omit important information, e.g. for the sentence "Faust made a deal with the devil", TextRunner would extract "Faust, made, a deal" instead of "Faust, made a deal with, the devil". These can partly be captured by syntactic constraints; however, this may cause the extraction of overly-specific relations such as "is offering only modest greenhouse gas reduction targets at". To tackle this, a lexical constraint is introduced. A relation has to appear with at least 20 distinct arguments in a sentence in order to be meaningful.

Although open information extraction is a promising research paradigm and it is possible to map clusters of relations to extraction schemas afterwards, it also provides an unneccessary restriction for the task of knowledge base population. Problems that ongoing research is addressing such as incoherent and uninformative extractions are issues which are less pronounced for bootstrapping methods, as introduced in the previous section.

2.3.3 Distant Supervision Approaches

Distant supervision is a method for automatically annotating training data using existing relations in knowledge bases. The first approach was proposed by Craven et al. (1999) in 1999 as a method for knowledge base population for the biomedical domain, and was called "weakly labeled". Although results were promising, this approach did not gain popularity until 10 years later, when Mintz et al. (2009) coined the term "distant supervision". The re-surfacing of these approaches may be partly due to the increasing availability of large knowledge bases on the Web. Mintz et al. (2009) define the distant supervision assumption as:

If two entities participate in a relation, any sentence that contains those two entities might express that relation.

How such an approach works in practice is visualised in Figure 2.3. The input to the approach is a knowledge base, containing a set of classes and relations, instances of those classes and examples of those relations, and training and test corpora. The training corpus is preprocessed to recognise named entities, then searched for sentences containing both the subject and the object of known relations (e.g. "Virginia" and "Richmond" for the relation "contains(LOC, LOC)"). Sentences containing both the subject and the object of known relations are considered positive training data for the relation, others are negative training examples (NIL). A supervised classifier (e.g. Naive Bayes, SVM, MaxEnt) is then trained and applied to a test corpus. Overall, the learning process is the same as that for supervised learning, merely the training data labelling process is different (automatic instead of manual). As such, the approach has all the advantages of supervised learning (high precision extraction, output with respect to extraction schema), and additional advantages since no manual effort is required to label training data. Extraction performance is slightly lower than for supervised approaches due to incorrectly labelled training examples. Improving the automatic labelling process has been the main focus of distant supervision research since, as a survey by Roth et al. (2013) outlines.

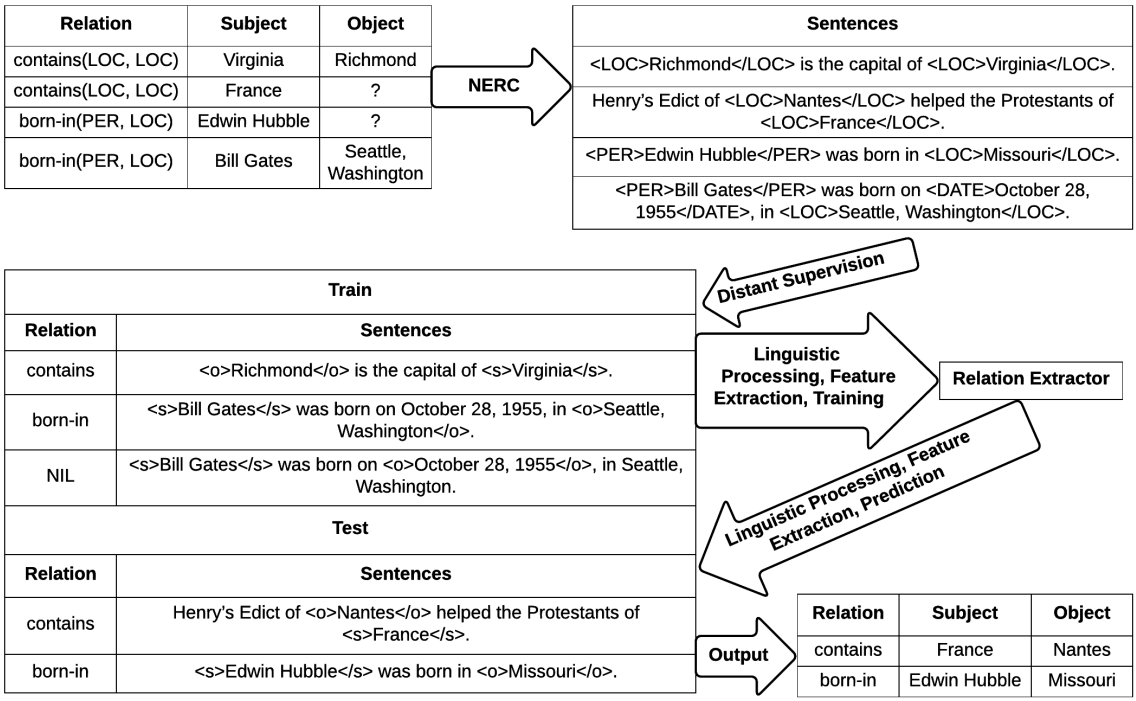

Figure 2.3: Mintz et al. (2009) Distant Supervision Method Overview

2.3.4 Summary

Table 2.1 summarises the key points of the three types of minimally supervised approaches. All three different relation extraction streams have different advantages and disadvantages. They differ with respect to how much initial input is required, if human intervention is required during the learning process and how suitable they are for knowledge base population. Semi-supervised / bootstrapping methods may only need a handful of initial examples, but as discussed in Section 2.3.1, the problem of semantic drift may require additional human intervention during the learning process. They are suitable for knowledge base population as extraction is performed with respect to an extraction schema. Unsupervised / Open IE approaches, on the other hand, do not require any input to start with. This means, however, that the output of such approaches is merely clusters of relations and there is no straightforward way of mapping them to an existing relation schema. Therefore, they are interesting for scenarios for which such an extraction schema is not available or for which the goal is to extend an extraction schema, but they are less suitable for knowledge base population. Lastly, distant supervision approaches require the most input out of the three methods initially, around 30 examples per relation at least. The abundance of such information on the Web in existing knowledge bases (see Section 2.2.3) makes it possible to gather such information automatically and as such, they do not require human input. Because they then also use the schema associated with relation examples for training, they are very suitable for knowledge base population. Based on the analysis presented in this section, distant supervision was picked as the most suitable relation extraction approach given the research scope of this thesis. Note though that this does not mean distant supervision is the best possible relation extraction approach overall. If large quantities of manually labelled training instances are available, those might be more preferable than automatically labelled data. The next section now discusses the state of the art in distant supervision in more detail.

Method	Semi-supervised	Unsupervised	Distantly supervised
Input	Unlabelled text, extraction schema, rules and/or examples	Unlabelled text	Unlabelled text, extraction schema, examples
Output	Extraction rules, relations	Groups of relations	Supervised classifier, relations
Description	Using a small set of extraction rules, extract examples, keep prominent ones, iteratively learn more extraction rules and examples	Discover groups of relations from text using clustering, keep prominent ones	Using a schema and examples of extractions, automatically annotate training data, train a classifier to extract more data
Advantages	Easy to add new rules, can also be supplied by user	No knowledge about text necessary	Extractions with high precision and recall
Disadvantages	Often low recall and/or manual refinement needed for high precision	Difficult to make sense of groups and map to extraction schemas	Initial examples required

Table 2.1: Comparison of different minimally supervised relation extraction methods

2.4 Distant Supervision for Relation Extraction

As introduced in Section 2.3.3, distant supervision is a relation extraction method which exploits existing examples in a knowledge base for automatically labelling training data. In this section, a detailed comparison of research on distant supervision is performed with regard to the key aspects in which they differ: what kind of background knowledge and corpora they use (Section 2.4.1), how automatic labelling is performed (Section 2.4.3), how named entity recognition and classification for distant supervision is performed (Section 2.4.4), how relations are extracted and evaluated (Section 2.4.2), and, lastly, what applications of distant supervision methods exist (Section 2.4.5). Limitations of the state of the art in distant supervision are explained in Section 2.5 and the main findings are summarised in Section 2.6.

2.4.1 Background Knowledge and Corpora

As mentioned in the last section, distant supervision approaches rely on a corpus and a background knowledge base to automatically label sentences in a corpus for training. Early approaches to distant supervision use a variety of different corpora and mostly small domain knowledge bases. Craven et al. (1999) use the Yeast Protein Database (Hodges et al., 1998) and match it to abstracts of PubMed[6] papers. Bellare and McCallum (2007) use BibTex citations and match them to the Cora data set[7] containing computer science research papers. Wu and Weld (2007) and Wu and Weld (2008) then extract information from Wikipedia and treat Wikipedia infoboxes as the corresponding knowledge base. Some later approaches also exist for domain-specific extraction, e.g. Roller and Stevenson (2014) use the biomedical knowledge base UMLS and the Medline corpus[8].

Most subsequent approaches, however, use large *cross-domain knowledge bases*. The most commonly used knowledge base is Freebase (Bollacker et al., 2008), used by Mintz et al. (2009); Riedel et al. (2010); Hoffmann et al. (2011); Surdeanu et al. (2012) and others; some research (e.g. Nguyen and Moschitti (2011a)) uses YAGO (Suchanek et al., 2008).

Freebase is a collaboratively created knowledge base, which has an extremely rich set of entities and relations, containing around 3 billion triples. Entities are organised in topics (e.g. music, book, media, people), which have associated classes (e.g. musical artist, politician). Entities are instances of one or more classes. Information in Freebase follows Semantic Web standards in that entities have unique dereferencable URIs and there are outward links to other data source. Knowledge in Freebase is partly imported from other sources[9], e.g. MusicBrainz (Swartz, 2002) or Wikipedia, the schema as well as the data are then edited by collaborators. The richness of the schema and the fact that data is edited to ensure higher quality and reduce redundancy make Freebase very suitable for distant supervision. As of April 2015, Freebase is being discontinued and transitioned to Wikidata (Vrandečić and Krötzsch, 2014).

The knowledge bases Wikidata, DBpedia (Bizer et al., 2009) and YAGO are all primarily

[6] http://www.ncbi.nlm.nih.gov/pubmed
[7] http://people.cs.umass.edu/~mccallum/data.html
[8] http://mbr.nlm.nih.gov/Download/
[9] http://wiki.freebase.com/wiki/Data_sources

based on Wikipedia and are thus multilingual. While DBpedia and YAGO are knowledge bases automatically constructed from Wikipedia, Wikidata is a collaboratively constructed knowledge base. The difference between DBpedia and YAGO is that, while DBpedia automatically converts Wikipedia infoboxes into a knowledge base and then interlinks it with other Linked datasets, YAGO re-uses WordNet for constructing its schema. YAGO2 (Hoffart et al., 2011) further uses hand-crafted patterns to reconcile duplicate relations, resulting in 100 relations, whereas DBpedia has many redundant and inconsistent relations. YAGO2 also normalises times and dates, which further facilitates automatic reuse. These additional efforts means YAGO and YAGO2 are potentially more suitable for information extraction purposes than DBPedia. YAGO2 consists of 120 million facts and is thus much smaller than Freebase; however, it has the advantages of integration with WordNet and time and date normalisation.

In addition to evaluation with Linked datasets as background knowledge, distant supervision has also been used as an approach to tackle the TAC KBP challenges (Surdeanu et al., 2010; Angeli et al., 2014a; Roth et al., 2012, 2014) and has thus been evaluated on TAC KBP corpora using locally defined extraction templates. The system by Roth et al. (2014) even won the KBP challenge in 2014, which further strengthens the argument of distant supervision being a suitable approach for knowledge base population (Section 2.3.4).

So far, YAGO has been used as a background knowledge base, but neither DBpedia nor Wikidata have been used. With Freebase being discontinued, this could possibly change[10].

In terms of different corpora, as already mentioned above, PubMed abstracts and the Medline corpus have been used for the biomedical domain. Approaches evaluated in a cross-domain scenario mostly use Wikipedia (e.g. Yao et al. (2010); Takamatsu et al. (2012); Ling and Weld (2012)) or the New York Times corpus (e.g. Riedel et al. (2010); Yao et al. (2010); Hoffmann et al. (2011); Ling and Weld (2012)). Recent approaches have also used a Web-based approach for acquiring training data (Dong et al., 2014; Vlachos and Clark, 2014b), similar to semi-supervised Web-based approaches (Etzioni et al., 2004; Carlson et al., 2010a): Web pages are retrieved with queries containing the subject of a relation and the relation name, hoping that those pages then contain the object of the relation. However, while those approaches make use of text on Web pages, they do not make use of semi-structured content such as lists or tables like NELL (Carlson et al., 2010a) does, nor of HTML markup like Wikipedia-based approaches do (Wu and Weld, 2008; Ling and Weld, 2012).

2.4.2 Extraction and Evaluation of Distant Supervision Methods

Distant supervision methods differ with respect to whether they perform sentence-level extraction or if they combine extractions for knowledge base population. Traditionally in the information extraction area, sentence-level extraction is prefered as it provides an estimation of performance independent of the number of or dependency between extractions. Gold standards such as the ACE corpus[11] are annotated with entities and relations, and the task is to use one part of the corpus for training and reproduce the results on a held-out (test) part of the corpus. Measures

[10] https://plus.google.com/109936836907132434202/posts/3aYFVNf92A1
[11] https://catalog.ldc.upenn.edu/LDC2006T06

such as precision, recall and F1-measure can then be computed on the corpus to estimate how exact the extraction method is (precision) and what the ratio of true positive extractions to all extraction is (recall). However, producing such a gold standard is very expensive since much effort goes into finding suitable annotators and merging annotations by different annotators. In order to still evaluate performance, distant supervision approaches use the following evaluation methods: 1) only annotating the most highly ranked extractions manually, 2) trying to map labels obtained through distant supervision to gold standards, or 3) not performing sentence-level evaluation, but entity-level evaluation instead.

For 1), test data is annotated using the same heuristic as for annotating training data, i.e. the distant supervision heuristic (Section 2.3.3). Sentences are then classified and ranked by confidence value of the relation classifier. The top ranked 1000 sentences are then selected and annotated manually. Mintz et al. (2009) perform such an evaluation as well as Hoffmann et al. (2011); Alfonseca et al. (2012); Ling and Weld (2012); Liu et al. (2014). 2) is something that is explored by Roller and Stevenson (2014). They make use of annotated corpora by identifying relations similar to the ones in their background knowledge base, UMLS. They semi-automatically map relations by measuring the overlap of $< s, o >$ relation tuples in UMLS with those of relations in gold standard corpora. They then manually examine the result and select suitable relations, discarding the very general "is-a" relationship. Mapping relations in the knowledge base is also explored by approaches which aim to combine both manually labelled and automatically labelled data for training (Sterckx et al., 2014; Angeli et al., 2014b; Pershina et al., 2014; Nguyen and Moschitti, 2011b).

The most popular evaluation setting is to not perform sentences-level evaluation, but entity-level evaluation (sometimes also called "aggregate evaluation"), or perform entity-level evaluation in addition to a sentence-level evaluation of the most highly ranked examples (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Alfonseca et al., 2012). The benefit of instance-level evaluation is that no manual effort is necessary to produce test data. For instance-level evaluation, an extraction is defined as correct if it also appears in the knowledge base. The number of all possible extractions is the number of entries for a relation in the knowledge base. This can also be seen as a task of reproducing part of the knowledge base. The problem, however, is that recall is defined differently and it is not possible to measure how many single extractions are missed. To combine the benefits of both sentence- and instance-level extraction, most studies report both, but sentence-level performance only for the most highly ranked ones, as mentioned above.

2.4.3 Distant Supervision Assumption and Heuristic Labelling

Early approaches to distant supervision use the assumption that every sentence (e.g. "Bill Gates founded Microsoft") which contains the subject and object of a relation contained in a background knowledge base (e.g. "Bill Gates" and "Microsoft" for "founder-of(PER, ORG")) also expresses that relation (Mintz et al., 2009). However, this heuristic can fail and generate *false positive* or *false negative* training data. False positives are created if the subject and object of a relation matches, but the sentence does not express that relation, e.g. for the sentence "Bill Gates spoke about Microsoft". False negatives occur if a sentence expresses a true relation which is not contained in

the knowledge base and is therefore considered negative training data (Min et al., 2013).

As a result, a number of research papers have focused on reducing such noise due to incorrect labelling, some of which have also been summarised in a recent survey paper (Roth et al., 2013). This section now gives an overview of such noise reduction methods grouped by type of approach.

At-least-one Models For Reducing False Positives

The distant supervision assumption is that all sentences which contain a known relation might be true positives. In practice, all such sentences are considered positive training examples for the respective known relation. As explained above this assumption does not always hold true. At-least one models therefore make a relaxed distant supervision assumption (Riedel et al., 2010), which differs from Mintz et al. (2009)'s assumption (see Section 2.3.3):

> If two entities participate in a relation, at least one sentence that mentions these two entities might express that relation.

Note that the wording of the "at least one" 'assumption is a bit confusing. What is meant is that at least one sentence is a true positive, which differs from the original assumption that all sentences are potential true positives. As an example, the assumption is that, given the relation "founder-of(Bill Gates, Microsoft')' at least one sentence that contains the entity pair "<Bill Gates, Microsoft>" expresses the relation "founder-of(PER, ORG)". This is in contrast to the original assumption, which is that all such sentences express the relation.

This at-least-one assumption is then added to the model as a constraint (Riedel et al., 2010; Yao et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Roth and Klakow, 2013b). Typically, the constraint is included in a graphical model over which inference is performed to estimate how likely a training example is to be a true positive training example. Approaches implementing this differ with respect to at what stage the constraint is used and how inference is performed. Riedel et al. (2010) were the first to propose an at-least one model. In a factor graph, an undirected graphical model, two tasks are modelled jointly: the task of predicting relations between entities and the task of predicting which sentences express these relations. This task is seen as a constraint-driven semi-supervised learning problem (Chang et al., 2008). The model does not have the information whether sentences indeed express the relation, so instead a constraint is applied which penalises violations of the at-least-one contraint. To find the most likely configuration of relation and relation mention varibles, Gibbs sampling with SampleRank (Wick et al., 2009) is performed. Hoffmann et al. (2011) then introduce MultiR, a multi-label extension to the multi-class at-least-one model introduced in Riedel et al. (2010). This extension addresses the issue of overlapping relations, e.g. the relation "employee-of" overlaps with the relation "ceo-of". The model is trained with a simple perceptron training scheme. MIMLRE (Surdeanu et al., 2012) is a further extension to MultiR, it is a jointly trained two-stage classification model. On the first layer, multi-class predictions for contexts are made, which are then used by the second layer, which is a collection of binary relation classifiers. The at-least-one assumption is a feature of the relation classifiers. In an evaluation on the Riedel et al. (2010) corpus, MIMILRE outperforms all previous methods, including Hoffmann et al. (2011).

Pattern-Based Models For Reducing False Positives

Alfonseca et al. (2012) propose the use of hierarchical topic models to classify the context of relations. Topics are modelled as patterns, using either the syntactic dependency path between entities, or simply the words between the two entities. They construct four different topic models, estimated with Gibbs sampling (Geman and Geman, 1984): a model which captures general patterns that appear with all relations, a model with captures patterns that are specific for an entity pair, but not typical for the relation, and finally a model which contains patterns that are observed across most pairs with that relation. The latter topic model is used to estimate $P(r|w)$, the probability that a pattern w conveys a relation r. The results are not directly compared to previous approaches; rather, they are compared to an internal baseline: $p(r|w)$ based on a maximum likelihood estimate of the number of times that a pattern w has been seen connecting the two entities for which r holds divided by the total frequency of the pattern. There is improvement between using the topic models and using simple patterns for all four relations used in the evaluation, but no absolute numbers are given (precision is given with respect to a confidence value, recall is not given) and the results are difficult to interpret given the relatively simple baseline.

Takamatsu et al. (2012) also propose a pattern-based model for modelling whether a pattern expresses a relation or not. Compared to Alfonseca et al. (2012), they model more directly whether a pattern can express a relation or not. Compared to at-least-one models, the approach does not fail if an entity pair is only mentioned once in a corpus (Riedel et al., 2010). The idea of the generative model is that contexts either express a relation, or have many $< s, o >$ tuples that have an overlap with other patterns expressing that relation. Pattern co-occurrence probabilities are computed, followed by an inference process based on the overlap of $< s, o >$ tuples. A probabilistic graphical model is learned with a hidden variable to model if a pattern expresses a relation. One of the main differences between the model of Alfonseca et al. (2012) and that of Takamatsu et al. (2012) is that Alfonseca et al. (2012) do not group occurrences of contexts for all $< s, o >$ tuples. The model is evaluated on Wikipedia, compared against Mintz et al. (2009) and Surdeanu et al. (2012) and outperforms both approaches.

Modelling Missing Data For Reducing False Negatives

A further group of methods address the problem of *false negative* training data, which arises if a training example which should be marked as positive is marked as negative for a relation because the example is missing from the background knowledge base. This happens surprisingly often, even in large knowledge bases. Min et al. (2013) show that, as of 2013, Freebase was missing a nationality for 78.5% of all persons.

One possible approach followed by Min et al. (2013) is to only make minimal use of negative training data by learning from almost exclusively positive training data and unlabeled data. Their model is an extension to the MIML model (Surdeanu et al., 2012) and is a 4-layer hierarchical model with positive and unlabeled data and a small sample of negative data as input. Latent variables are used to model the true labels for the training instances, which are then the input to MIML. The model is trained iteratively with an expectation maximisation algorithm and log-likelihood as objective function. Full inference over the search space is performed. An evaluation

on the Riedel et al. (2010) dataset shows that the proposed approach is a small improvement over the approach it extends (Surdeanu et al., 2012). Moreover, the authors suggest that such an approach could easily be incorporated into any distant supervision approach. While this might work, the approach requires full inference over the search space and is thus very expensive, but only brings small improvements.

Ritter et al. (2013) also propose to model missing information with latent variables, but in contrast to Min et al. (2013) they only perform local search instead of inferencing over the full search space and their method is thus more suited for large datasets. They build their model on top of MultiR (Hoffmann et al., 2011). In addition, they incorporate the popularity of entities in Freebase and add a manually set parameter for the popularity of relations. The latter is to distinguish between relations with the same types as arguments, e.g. "contains(UK, London)" vs "capital_of(UK, London)". The parameter is set so that relations with a greater true positive rate are preferred over relations with lower true positive rate. Results of the model without popularity give an improvement over the model it is an extension of, Hoffmann et al. (2011), and are similar to another method for reducing false negatives by Xu et al. (2013). Incorporating popularity of entities and relations gives a bigger boost. However, the latter relies on setting parameters manually. Furthermore, it is unclear if the approach would also improve results with MIML as a base model.

Xu et al. (2013) propose a method based on pseudo-relevance feedback. The benefit of their method is that relation labels are corrected before training the relation extractor, thus no inference is needed. Sentences are first annotated with relations, then a passage retrieval model is learned to provide relevance feedback on the annotated sententences. The passage retrieval model is based on the idea that entity pairs that appear in more sentences and more relevant sentences are more likely to express the relation. Relevant sentences are sentences which contain a positive training example, others containing negative training examples are irrelevant sentences. The model uses coarse features, such as bag of words features for high recall, and is trained with an SVM. After training, the model is applied to all testing sentences, and relevant sentences are used as training data for MultiR. The resulting approach shows encouraging results, outperforming MultiR in terms of recall at the same precision. Compared to Min et al. (2013) and Ritter et al. (2013) it is inexpensive and could easily be integrated with other approaches that, e.g. address the true positives problem.

Universal Schemas

The idea of universal schemas (Riedel et al., 2013) is somewhat similar in spirit to methods for reducing false negatives. Methods for modelling missing data to reduce false negatives assume that not all relation mentions (e.g. "Microsoft founded-by Bill Gates") are contained in the KB, which leads to them being labelled as negative training data. Universal schemas, on the other hand, address the idea that not all relations (e.g. "founded-by") are contained in the KB. They then aim at combining relation mentions extracted with distant supervision with respect to the Freebase schema with other relations, e.g. with relation mentions discovered in text with Open IE methods. Recall that Open IE methods (Section 2.3.2) do not rely on an extraction schema, and

instead cluster surface patterns (e.g. "founded", "was founded by") to relations (e.g. "founded'-by'). Universal schemas avoid the need for just using one relation schema (e.g. Freebase) to extract relations against and instead provide a union of extracted relations for several schemas (Freebase, Open IE clusters). One benefit of universal schemas is that they can be used for textual reasoning, e.g. "A ceo-of B" implies "A employee-of B", which is something that existing distant supervision approaches also do not address.

The problem is modelled as a matrix factorisation problem with entity-entity pairs (e.g. <Microsoft, Bill Gates>, <Larry Page, Google>) in the rows and relations in the columns (e.g. founded-by, employee-of, ceo-of). The relations in the columns are relations defined by different schemas, e.g. Freebase or Open IE. The input is an incomplete matrix with values of 1 for known relation tuples and missing values for unknown relation tuples, the output is a completed matrix. For the missing values, a probability is learned using the logistic function and a natural parameter, capturing the compatibility between a relation (e.g. "founded-by") and a tuple (e.g. <Microsoft, Bill Gates>) based on the dot product of their latent feature representations. The confidence of a relation triple (e.g. <Microsoft, founded-by, Bill Gates>) to be true is assessed using a combination of learned collaborative filtering models capturing e.g. compatibility between NE types of the arguments and the relation or compatibility between the latent feature representations between a relation and a tuple.

The approach is tested on the Riedel et al. (2010) New York Times data; the top 1000 results are evaluated. Compared to MIML, they improve by 18 points in average precision and compared to Mintz et al. (2009) by 31 points. The results show the importance of considering integrating several extraction schemas, which is often neglected in relation extraction.

Adding Manually Labelled Data

Finally, some works propose to enrich manually labelled data with automatically labelled data, which demonstrates the usefulness of distant supervision for other tasks. Nguyen and Moschitti (2011b) show a Wikipedia-based distant supervision approach enriched with manually annotated ACE data, which outperforms an approach trained on ACE data alone. For this, they use YAGO as a background knowledge base and manually map YAGO relations to ACE relations. Two REs, one using distantly labeled data and one using ACE data, are trained based on kernel methods and the probabilities of those classifiers are combined linearly. Their approach outperforms an approach trained on ACE data alone by 3 points in F1.

Pershina et al. (2014) use KBP data as manually labeled data and implement an approach on top of MIML. Instead of training an RE on both datasets as Nguyen and Moschitti (2011b) do, they use the KBP data to inform the MIML training. On the KBP data, they learn guidelines for particular relations, consisting of a pair of semantic types for the two entities, and a dependency path, optionally lexicalised. Those guidelines are then incorporated into the MIML training. The approach is evaluated on the Riedel et al. (2010) corpus, and shows an improvement of 4 points in F1 over MIML and 6 points in F1 over a model trained on the KBP data alone.

Angeli et al. (2014b) propose to use active learning. The idea is to manually correct sample training instances iteratively which are likely to be useful, but also likely false positive candidates.

In practice 1% of all training instances are inspected. This is achieved by using three active learning criteria, sampling uniformity, and two criteria based on disagreement. The approach is evaluated on the KBP 2013 data and is based on the MIML distant supervision approach. Results show an improvement of 4 points in F1 compared to an approach without active learning. Although these results are promising, from an application perspective, the prior informed model by Pershina et al. (2014) or the linear combination of a supervised and distantly supervised model proposed by Nguyen and Moschitti (2011b) are more useful, since they reuse existing gold standards instead of requiring additional manual effort.

2.4.4 Named Entity Recognition for Distant Supervision

Named Entity Recognition and Classification is typically seen as a preprocessing step for relation extraction by distant supervision approaches. Almost all publications report using Stanford NERC (Finkel et al., 2005), others use a Wikipedia-based NER (Nguyen and Moschitti, 2011b).

Some research has been done on improving distant supervision by using fine-grained instead of coarse-grained Wikipedia-based named entity classifiers (Ling and Weld, 2012; Liu et al., 2014).

FIGER (Ling and Weld, 2012) is a Wikipedia-based fine-grained NERC system. The tag set for FIGER is made up of 112 types derived from Freebase, by selecting the most frequent types and merging too fine-grained types. The goal is to perform multi-class multi-label classification, i.e. each sequence of words is assigned one or several of multiple types or no type. Training data for FIGER is created by exploiting the anchor text of entity mentions annotated in Wikipedia, i.e. for each sequence of words in a sentence, the sequence is automatically mapped to a set of Freebase types and used as positive training data for those types. The system is trained using a two step process: training a CRF model for named entity boundary recognition, then an adapted perceptron algorithm for named entity classification. Typically, a CRF model would be used for doing both at once (Finkel et al., 2005), but this is avoided here due to the large set of NE types. An evaluation is performed in which MultiR (Hoffmann et al., 2011), a state of the art multi-label multi-class distant supervision system, is augmented with FIGER's NE types. This results in 224 additional features representing binary indicators of NE types, which are simply aggregated with each relation extraction feature vector. The system is evaluated using 36 relation types from the NELL knowledge base (Carlson et al., 2010a) and the NYT corpus (Sandhaus, 2008). Results show that MultiR+FIGER achieve a maximum F1 of 0.4, compared to the original MultiR achieving 0.207 on the same corpus. The highest improvements are for relations for which there is no straightforward way of representing them with the traditional 3 NE types (PER, LOC, ORG), e.g. teamPlaysIn-League(Sports_team, Sports_league) or musicianInMusicArtist(musician, musicArtist).

Liu et al. (2014) then aim to improve on the method presented in (Ling and Weld, 2012) by exploring different methods for exploiting fine-grained NER information which go beyond simple aggregation. For each NE in the test data, their types are predicted by retrieving the top 20 Bing search snippets and then tagging each mention of the NE in the search snippets with a Wikipedia-based fine-grained NER using the FIGER tag set trained in a similar fashion. Final types for each NE are predicted by obtaining a ranked list of types for each NE and sorting them by prediction

scores. The fine-grained NERC is then integrated with MultiR, and three different methods of combining the NE types with RE features are tested. The first method substitutes coarse Stanford NERC types with fine-grained NE types, but also adds super-types of those fine-grained NE types, e.g for "/person/politician", the super-type "/person" would also be added. The second method is to augment the Stanford NERC type features with fine-grained NERC features. The third method is to only add fine-grained NERC features for sparse feature vectors with fewer than 30 features. For evaluating their approach, the NYT data is used, using part of it as held-out for testing as in Riedel et al. (2010); Hoffmann et al. (2011). Evaluation is performed both on instance-level and on sentence-level, using the manually labelled 1000 sentences from (Hoffmann et al., 2011) for the latter. Out of the different NER aggregation methods, the third method of adding fine-grained NERC features only for sparse feature vectors performs best. Instance-level results compared with MultiR only show marginal improvements; however, sentence-level results show a significantly higher precision at all points of recall. No direct comparison to Ling and Weld (2012) is made and no F1 results are given, but the improvement is much smaller than that reported by Ling and Weld (2012), either due to the slightly different evaluation method or the Bing expansion method.

2.4.5 Applications of Distant Supervision

Several approaches which apply the idea of distant supervision to solve NLP tasks other than relation extraction have been proposed, which demonstrate the usefulness of distant supervision beyond relation extraction. Note that the term "distant supervision" is used very loosely for some of those applications.

Marchetti-Bowick and Chambers (2012) identify keywords related to political subtopics (e.g. Obama, Ideology) and make the assumption that if that keyword occurs in that tweet, it is about that topic. In addition, they use sentiment words to identify the sentiment of each tweet. The combination of the two, topic keyword and sentiment word, leads to an aspect-based sentiment analysis approach. Unlike distantly supervised relation extraction aproaches, they do not use tweets containing both words for training one classifier, but first train a topic classifier, then train a second classifier for sentiment on topic-relevant tweets. For both stages, a multinomial Naive Bayes classifier is used. For topic identification, findings are that for a Twitter corpus only containing political tweets, a very high F1 score of around 90% can be achieved with such an approach, but for general tweets, only an F1 score of around 18% can be achieved, mostly due to precision being around 10%. At the second stage, sentiment classification, it is shown that the aspect-based sentiment analysis appraoch with distant supervision outperforms a lexicon-based approach. While the reported results show the benefit of the distant supervision idea, the authors do not discuss why they opted for a two-stage classification approach instead of labelling tweets with both arguments, then training a classifier, as for distantly supervised relation extraction. It would be interesting to see how this would compare to the two-stage setting.

Exner et al. (2015) propose to use distant supervision creating semantic role labelling resources in languages other than English, then training a semantic role labeler on those resources. They start with the English version of PropBank and English Wikipedia. The goal is then to identify

propositions in the Swedish Wikipedia, e.g. they try to translate the English predicate "win.01" to Swedish predicate "vinna.01". Using Wikipedia disambiguation pages and external named entity linking tools, they map mentions of NEs to unique Wikipedia-based identifiers in both English and Swedish Wikipedia. They then identify propositions in the English version of Wikipedia and for sentences which contain them, get the unique Wikipedia-based identifiers. The distant supervision idea of the approach is to then use those pairs of named entities linked to identifiers with a proposition to identify pairs in the Swedish Wikipedia. Because they use the same Wikipedia-based identifiers for NEs via cross-language links, the propositions can be transferred from one corpus onto the other corpus in a straight-foward way, by identifying sentences which contain the same pairs of identifiers. The sentences with automatically aligned propositions are then used to train a semantic role labeler. The overall idea of the approach is very similar to distant supervision for relation extraction, with the difference that the corpus, in that case Swedish Wikipedia, is not directly annotated with the background knowledge base, but indirectly via a semi-parallel corpus, English Wikipedia. This makes the task more challenging than distantly supervised relation extraction. However, they still report reasonably high results – a precision of 58% at a recall of 47%.

Parikh et al. (2015) experiment with using the distant supervision idea to train a semantic parser. Whereas relation extraction focuses on extracting binary relations, semantic parsers learn to recognise additional semantic relations such as "cause" or "theme". Thus this can be seen as a more complex form of knowledge extraction, where the event structure has the representation of a semantic parse. In the case of events, it is very difficult to find all arguments of events in one sentence. Instead, they decompose the events into subevents, then later augment the local events. They evaluate on the GENIA event extraction shared task data (Kim et al., 2009), on which they outperform 19 of 24 submissions. The distantly supervised approach alone achieves a precision of 29.4% at recall of 19.1%. What brings big improvements is collecting five trigger words for each event and incorporating them into learning, which radically improves results to a precision of 72.2% at a recall of 27.9%. It is interesting to see that trigger words bring such a big improvement for distantly supervised event extraction. Maybe this is something that could also bring improvements to distantly supervised binary relation extraction, i.e. to manually define trigger words for relations.

Magdy et al. (2015) use the idea of distant supervision and apply it to classifying tweets into topics with the help of YouTube labels. They do so by collecting tweets which contain links to YouTube videos and then retrieving the topic the video is assigned, which is one of 18 coarse-grained classes. They merge those to 14 classes, thereby avoiding too sparse or too general classes, and end up with categories such as "Pets & Animals". A topic classifier is then trained on such tweets containing YouTube links and can be applied to other tweets which do not have to contain YouTube links. In practice they perform a hold-out experiment. The overall idea of using distant supervision is similar to that of Marchetti-Bowick and Chambers (2012), apart from that they do not train a sentiment classifier afterwards. They do not test their approach on general tweets as Marchetti-Bowick and Chambers (2012) do, instead only tweets which they already know to contain one of the topics. As such, their results are relatively high, around 57% precision and

recall, but it is unclear how the approach would perform in a less controlled setting.

Plank et al. (2014) also exploit links in tweets for part of speech tagging of tweets, but instead of restricting themselves to particular websites and collecting labels, they use links to retrieve richer linguistic information from those websites that are linked. Crucially, linked websites are only used during training, but not required during testing. What happens during training is that words appearing in the tweet are aligned with words on linked websites, so that more context for those words is available. The tag most frequently assigned to those words on the website is then projected to the occurrence of the word in the tweet. They call their method "not-so-distant supervision" and indeed, the method is only vaguely related to distantly supervised relation extraction. The general method of acquiring additional information from linked websites is a strategy also used for Twitter-based entity linking (Gorrell et al., 2015) and could also be used for relation extraction from social media or Web data.

Fan et al. (2015) propose to use a variant of distant supervision with Freebase for entity linking. Entity linking approaches are typically trained with Wikipedia. In Wikipedia, text is annotated with links, most of which are named entities. These annotations can then be used to train models for entity linking. The idea of Fan et al. (2015) is to achieve something similar with Freebase instead of Wikipedia. To do so, they make use of the Freebase property `/common/topic/topic_equivalent_webpage` to collect Web pages which are known to be about specific entities. Whenever they find an entity's name on those Web pages, they then annotate them with their Freebase ID. This can be used to train an entity linking approach with Freebase as a background knowledge base and linked Web pages, in addition to Wikipedia pages as text. It would be interesting to see how useful only using the topic related Web pages would be for training distantly supervised relation extractors. In particular, the authors do not discuss how many Freebase entries have such linked Web pages, so it would be interesting to study how many entities do and if so, how useful they are for training relation extractors.

2.5 Limitations of Current Approaches

Most distant supervision approaches (Section 2.4.2) use the same distant supervision paradigm for creating training as well as test corpora, with the exception of Roller and Stevenson (2014), who try to map relations to existing gold standard corpora and then reuse those. What existing approaches do not evaluate is relation extraction across sentences using coreference resolution, as e.g. annotated in gold standards such as the ACE 2005 Multilingual Corpus. Further, approaches which perform instance-level extraction for knowledge base population focus on relation extraction, and leave validation of extractions, a popular task as part of the TAC KBP challenges, to future work.

Section 2.4.3 explains that the distant supervision assumption can cause incorrect labelling and summarises different methods for improving on that. At-least-one models make the assumption that at least one of the sentences in which an entity pair is mentioned is a positive training example. The suitability of the context for a relation is learned in concert with the suitability of an entity pair for a relation. However, this assumption can fail, then leading to low performance.

In addition, performing inference in the context of learning the graphical models can be very expensive. Pattern-based models do not have the at-least-one restriction. However, they still rely on expensive graphical models and in addition, they rely on expressing relations in terms of patterns. Approaches combining data labelled with the distant supervision assumption with manually labelled data is sensible for some use cases where such data already exists (e.g. for the TAC KBP challenges); however, for most scenenarios, additional manually annotated training data is not available. Approaches addressing the problem of the false negatives model do this by avoiding the use of negative training data; by incorporating latent variables and performing inference over the search space; or by using pseudo-relevance feedback. The best-performing approach of the ones discussed (Min et al., 2013) only shows minor improvements over MIML, an approach addressing the problem of false positives, but is computationally expensive. Moreover, the problem might be specific to the relations selected and also depends on how negative training data is selected. The research in this thesis therefore only focuses on the problem of false positives. Approaches combining distant supervision with supervised data using the ACE or KBP 2011 corpus show an improvement over both training distantly supervised RE models alone and over training supervised models alone. However, the reason for this is not uncovered, i.e. is it due to currently available hand-labeled RE corpora being small and still being able to benefit from more training data, even if it is noisy, or does the training data happen to be complementary? Future work still needs to uncover the relationship between size of manually labeled RE data and usefulness of additional automatically generated RE data.

Section 2.4.4 shows that most distant supervision methods use Stanford NERC for pre-processing. There are two methods which use fine-grained NERC for distant supervision (Ling and Weld, 2012; Liu et al., 2014) and show promising results. However, both of them rely on Wikipedia for generating training data by specifically exploiting the anchor text of entity mentions and Wikipedia categories to map to Freebase types. For NE types not annotated in such a resource, or for testing documents which are not very similar in style to Wikipedia articles and would thus be considered out of genre, this approach would not be suitable. Overall, research on fine-grained NERC for distant supervision shows promise, but still leaves much room for future work. Most importantly, existing distant supervision methods view NERC as a preprocessing step. Such a pipeline architecture can lead to errors made at an earlier stage of the pipeline (e.g. NERC) being propagated to a later stage of the pipeline (e.g. RE). Future work could focus on jointly learning models for those the tasks, thus learning dependencies between the stages.

2.6 Summary

Distant supervision, a relation extraction method that uses relations defined in a background knowledge base to automatically label training data, has become a popular research area since 2009. Research efforts have mostly focused on improving automatic labelling to reduce false positives and false negatives (Section 2.4.3), and there has been some work on improving NERC for distant supervision (Section 2.4.4), and on integrating distant supervision with Open IE (Riedel et al., 2013). Distant supervision approaches further differ with respect to what knowledge base

and corpus they use: most approaches use Freebase and either Wikipedia or the New York Times corpus, and a handful use the YAGO knowledge base or biomedical knowledge bases and biomedical corpora (Section 2.4.1). Distant supervision is either used for sentence-level or instance level extraction, and some approaches try to reuse gold standards as test corpora, whereas most perform a held out evaluation (Section 2.4.2). Applications of distant supervision include semantic role labelling, Twitter tagging, parsing, classifying YouTube labels and entity linking (Section 2.4.5), which further demonstrate the usefulness of distant supervision.

Chapter 3

Research Aims

3.1 Methodology Overview and Experiment Design

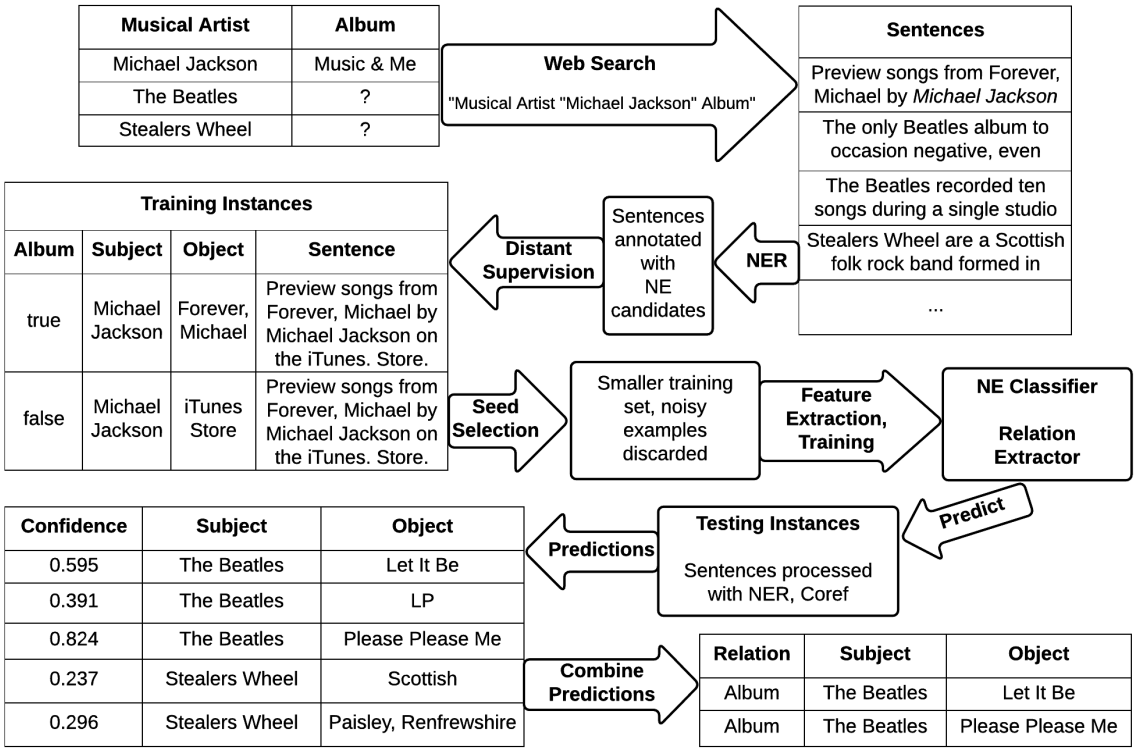

Figure 3.1: Overview of Distant Supervision Approach of this Thesis

This chapter explains the thesis aims and how the contributions of this thesis, for which experiments are described in subsequent chapters, fit together to advance the state of the art. The contributions, described in more detail in Section 1.2, either directly influence the state of the art in relation extraction with distant supervision or broaden the general understanding of named entity recognition for distant supervision. To guide this description, an overview of the distant supervision approach of this thesis is given. References to the related work discussion in

the previous chapter are made to explain research gaps and motivate the research aims of this thesis.

To recap, the contributions of this thesis concern:

- New statistical methods for selecting training instances for distant supervision
- An evaluation of methods for obtaining more testing instances for distant supervision via co-reference resolution methods
- Experiments on combining predictions of extracted relations
- The setting of the distant supervision approach as an entity-centric Web search-based relation extraction approach which gives instance-level results for knowledge base population
- A quantitative study on NERC in diverse genres, analysing reasons for NERC failure
- A new method for jointly training a NEC and a Relation Extractor for distant supervision with imitation learning
- Evaluating and comparing different methods against a distant supervision approach with imitation learning
- A study on how features extracted from HTML markup (e.g. links, text in bold or italics, or also lists) can help improve NERC performance

Figure 3.1 illustrates how the different methods documented in this thesis fit together. The approach starts with an initial incomplete knowledge base populated with entities and relations between some of them. The goal is to populate the knowledge base with more relations. The Web search-based setting provides training and testing instances for distant supervision. Training instances are the ones for which both subjects and objects of relations are known (e.g. <Michael Jackson, Music & Me>), whereas testing instances are those for which only the subjects of relations are known, but not the objects (e.g. <The Beatles, ?>). Named entities are then recognised and sentences are annotated with relations. Afterwards, some of those training instances are selected for training (Training Data Selection). Sentences are processed using a standard NLP pipeline including a sentence splitter, tokeniser, part of speech (POS) tagger and parser. From those sentences, features, including those based on POS tags, the parse tree and Web-based features are extracted. Extracted features are then used to train models for distantly supervised relation extraction. One option for this, proposed in the thesis, is to train NEC and RE models jointly with imitation learning. Testing instances are selected, which includes resolving co-references in testing sentences. The NEC and RE models are then applied to the testing sentences and relations are predicted for known subject entities.

3.2 Setting and Evaluation

3.2.1 Setting

Existing distantly supervised relation extraction approaches generally contain the following five components, also displayed in Figure 2.3 above: named entity recognition and classification; automatically labelling sentences with the distant supervision heuristic; preprocessing and feature extraction; training a classifier; and combining and returning results.

The approach described in this thesis adds to these components an additional component for training and testing data retrieval. The aim of the setting is an entity-centric Web search-based distant supervision approach which gives instance-level results for knowledge base population.

Most documented distant supervision approaches perform experiments on corpora such as the New York Times corpus or Wikipedia (see Section 2.4.1). They process each sentence with named entity recognisers. For those sentences that contain at least two entities, they iterate over each relation pair from a background knowledge base to label sentences with relations, which are then used as positive training examples for those relations. Negative training examples for relations are named entity pairs which are not identified as being in any relation in the knowledge base and are sampled randomly from the corpus.

Attempting to match each named entity in the corpus with each named entity in a background knowledge base is a significant computational effort. In reality entities and relations discussed in documents are on different topics, e.g. some documents are about politicians and their parties, whereas others are about musical artists and their albums. This means only a fraction of documents in a fixed corpus such as the New York Times are relevant to a specific entity in a background knowledge base. Therefore, attempting to match NEs tuples with every sentence in every document could lead to many false positives. A solution to this would be to preprocess the documents to find out if they are about the entity in question. However, a static corpus might not even contain the information desired, and significant effort could go into finding a suitable corpus which does.

Consequently, a different setting is proposed in this thesis (see Figure 3.1): instead of processing a static corpus and extracting all relations from it, the setting assumes that there is a user with a particular query, e.g. “What albums did the musical artist Michael Jackson release?”. The queries should contain the type of the subject entity (musical artist), its name (Michael Jackson) and the type of the object entity (album). The query is then used to retrieve sentences from the Web using a search engine. The search engine functions as a preprocessing step to retrieve relevant Web pages. Moreover, this is a dynamic way of retrieving information, rather than the static corpus-based way and could be used in a real-world setting where a user has a specific query and wants to retrieve the answer to such a query. This has the additional benefit of having access to large quantities of information, which eliminates the need for having to search for a suitable corpus that contains the desired information.

The same setting is used for training and testing. To generate annotated training data, Web pages potentially related to that query are then retrieved using a search engine, and all NEs on the Web pages matched against the named entity in the query (Michael Jackson) and objects of the relation “album_of” with the subject “Michael Jackson”, as already contained in the knowledge

base (Music & Me). Sentences which contain both the subject and the object of the relation are used as positive training data. NE pairs with the subject of the relation, but a different object, i.e. a mention of an entity of the type specified by the relation, but not one referring to an entity known to stand in the given relation to the subject entity, are used as negative training data. Note that surface forms such as "Michael Jackson" can refer to multiple real-life entities, which the task of named entity disambiguation is concerned with. This issue is left for future work.

3.2.2 Evaluation

In order to measure the performance of a distant supervision approach, a test set is neccessary. Information extraction approaches usually rely on gold standard corpora produced in the context of evaluation initiatives such as ACE (Walker et al., 2006) or Ontonotes (Hovy et al., 2006). Another possibility is to use benchmark data provided by evaluation challenges, e.g. TAC KBP Surdeanu and Ji (2014).

The problem is that, for relation extraction, not many large manually annotated corpora exist and it is particularly difficult to find gold standard testing corpora for the same genre as the training data. One approach for solving this is to find a gold standard which contains some of the desired relations, and then to only evaluate those relations which can automatically be mapped to that gold standard (Roller and Stevenson, 2014). Other existing distant supervision approaches use the same method for obtaining test data as they use for obtaining training data, i.e. automatically annotating it with relations from a knowledge base. In that case, part of the knowledge base is used for training, while another part is used for testing (also called "hold-out evaluation"). They then perform a sentence-level or an instance-level evaluation. For sentence-level evaluation (Mintz et al., 2009; Hoffmann et al., 2011; Alfonseca et al., 2012; Ling and Weld, 2012; Liu et al., 2014), testing instances are classified by a model and then ranked by confidence of each prediction. Top ranked sentences are then annotated manually. Another possibility is instance-level evaluation (Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Alfonseca et al., 2012) for which predictions for the same $< s, o >$ relation tuples are aggregated. A prediction is deemed correct if it is contained in the knowledge base. This evaluation setting is therefore a good measure for knowledge base population performance.

For the Web genre, no corpus of Web pages with manually annotated Freebase relations is available. The closest suitable corpora would be the TAC KBP challenge corpora, which consist of manually annotated Wikipedia corpora. However, Wikipedia is a curated text collection and articles have very similar structures. Therefore, Wikipedia text is not very diverse and not a good representation of the Web genre overall. An example of a large and diverse Web corpus is the ClueWeb corpus[1]. However, the only annotations which exist for it are unofficial automatic NE annotations provided by Google researchers[2].

One of the research aims is therefore to create a new corpus for the Web genre using the entity-centric search-based method proposal described in Section 3.2.1, which is made available publicly. Evaluations are be performed on instance-level, to measure performance for knowledge

[1] http://lemurproject.org/clueweb12/
[2] http://lemurproject.org/clueweb12/FACC1/

base population, and on sentence-level. For the latter, the highest ranked results are marked as correct or incorrect, following existing work.

3.3 Selecting Training Instances

While distant supervision is a successful method for creating training data for any domain for which a small number of extractions are already available, it also has its drawbacks. One of the biggest drawbacks of distant supervision is that the method of automatically annotating training data produces noise, which affects relation extraction performance. As explained in Section 2.4.3, the distant supervision heuristic produces some false positive and false negative training data. False positives are created if a relation tuple from the background knowledge base matches two named entities that appear in the same sentence, but the sentence does not express that relation. False negatives are created if relations are missing from the knowledge base and then used as negative training data.

There have been several research efforts to prevent distant supervision methods from producing noisy training data. Existing methods fall into the categories described in Section 2.4.3. One goal of the research described in this thesis is to focus on new methods for reducing false negative training samples.

One possibility that has been explored is to infuse heuristically labelled data with manually labelled data (Sterckx et al., 2014; Angeli et al., 2014b; Pershina et al., 2014; Nguyen and Moschitti, 2011b), either by adding manually labelled data directly or by iteratively improving the quality of training data using active learning (Sterckx et al., 2014; Angeli et al., 2014b). While these methods are successful, also on other domains than the ones reported in the publications, additional manual effort is neccessary.

Other possibilities are to change the relation extraction model, e.g. by using the "at-least-one" assumption. At-least-one models assume that at least one of the training examples is a true positive (see Section 2.4.3). They then include this assumption in their relation model, e.g. by using factor graphs and performing global inference (Riedel et al., 2010). Results for such methods are mostly positive, however, global inference is computationally very expensive.

Another possibility is to use a different model for preprocessing the training data, e.g. hierarchical topic models (Alfonseca et al., 2012; Roth and Klakow, 2013a), pattern correlations (Takamatsu et al., 2012), or reranking (Xu et al., 2013). The idea of those models is that the contexts of true positive examples are different from the contexts of false positive examples.

This thesis takes an alternative approach, which tries to assess the ambiguity of surface forms of relation mentions using more background knowledge from the knowledge base. The assumption is that surface forms which are very ambiguous are likely to lead to noisy training data. Training examples containing such ambiguous NE mentions can then be discarded before training. Such an approach is computationally inexpensive and could be combined with other preprocessing approaches focusing on detecting unsuitable contexts (Alfonseca et al., 2012; Roth and Klakow, 2013a; Takamatsu et al., 2012; Xu et al., 2013). Experiments for this approach are described in Chapter 4.

3.4 Named Entity Recognition

3.4.1 Named Entity Recognition of Diverse NEs

As already indicated in Section 3.2.1, one of the central other natural language processing tasks relation extraction relies on is named entity recognition and classification. Concretely, for distant supervision, before sentences are annotated with relation tuples from a knowledge base, all named entities in those sentences are identified. Candidates for positive and negative training data are those pairs of NEs which are both identified by a NERC tool such as Stanford NER (Finkel et al., 2005) and also contained in a background knowledge base as a relation tuple. It is therefore crucial for relation extraction performance that the NERC approach used for preprocessing has a high precision and recall.

While off-the-shelf NERC approaches exist for genres traditionally studied in the NLP community, such as the newswire genre, not as much training data is available for less well studied genres such as the Web or newly emerging ones such as the social Web. Applying NERCs trained for one genre to another genre causes a drop in performance, which has led to research on domain adaption and transfer learning (e.g. Daumé (2007); Arnold et al. (2008); Guo et al. (2009)).

Further, information extraction is more challenging for some genres than for others, as some genres, e.g. Web data, blogs, social media or chat are characterised by a large degree of noise (Subramaniam et al., 2009) such as grammar and spelling mistakes, and in general lexical variation, which leads to lower precision and recall than in the newswire genre (Subramaniam et al., 2009; Derczynski et al., 2013).

While this has been observed by several studies, there is no study which systematically analyses what the main reasons for NERC failure in diverse genres are. One of the goals of this thesis is to answer why NERC approaches perform poorly on diverse genres, such as the Web genre and social media, or in other words, why they have more problems generalising from training to testing data for diverse genres. To answer this, several benchmark corpora for different genres, including the Web genre, are analysed. Corpus statistics which quantify the diversity of a genre such as the ratio between NE and unique NEs are measured and compared across corpora. Experiments are performed using different NERC approaches, ranging from Stanford NER (Finkel et al., 2005)[3], which is typically used as a preprocessing approach for distant supervision, to methods which try to avoid the "unseen NE" problem by using word embeddings (Collobert et al., 2011). Experiments on this quantitative analysis of NERC generalisation are reported in Section 5.

Hypotheses which are tested are if this is due to more diverse genres containing a larger proportion of unseen NEs than less diverse genres or because the context is more diverse and therefore NERC approaches have to deal with unseen features. Moreover, it is analysed whether NERC performance on diverse corpora being lower than for corpora of traditionally studied genres is due to only small training corpora being available for such diverse genres. One solution which is commonly used if there is no large training corpus from the same genre is to instead train on a large corpus from a different genre. Experiments are performed to analyse for which genres such a strategy might be beneficial.

[3] http://nlp.stanford.edu/projects/project-ner.shtml

3.4.2 NERC for Distant Supervision

Lessons learnt from those experiments are then applied to research how the task of NERC can be improved for distant supervision for the Web genre. Currently there is only limited research in this area; most research for distant supervision focuses on reducing noise for heuristic labelling (which is also studied in this thesis, see Section 3.3).

Previous studies for improving NERC for distant supervision have made the hypothesis that the main problem is that Stanford NERC produces coarse-grained NE labels, which are not always a good fit for relation types. They therefore propose training a NERC with fine-grained NE types (Ling and Weld, 2012; Liu et al., 2014) using Wikipedia. However, such an approach only works if additional annotated NERC training data is available for that genre. Although their fine-grained NERC, FIGER, might also perform better than Stanford NERC for out-of-genre scenarios, a drop in performance would still be expected compared to using NERC training data for the same genre.

The aim of the research described in Chapter 6 is to jointly train a NERC and relation extractor using only the training data automatically annotated with the distant supervision assumption. Traditionally, NLP tasks use a pipeline architecture, where models for different parts of the pipeline (e.g. NEC, RE) are trained separately. However, this ignores the fact that there are dependencies between the different tasks. In addition, if an error is made at an early stage in the pipeline, it is propagated to a task at a later stage of the pipeline. Such errors can be reduced by jointly learning models for different stages, since then, dependencies between the different tasks are learned. Methods explored for this in the context of natural language processing are e.g. integer linear programming (Roth and Yih, 2004, 2007; Galanis et al., 2012) and markov logic networks (Domingos et al., 2008; Riedel et al., 2009). Ideally, all different possibilities of dependencies between the tasks would be explored by performing full inference over the search space. However, this is computationally very expensive. A cheaper method is to only explore parts of the search space which are likely to be relevant. One way of doing this is with imitation learning.

This joint approach proposed in this thesis therefore uses the structured prediction method imitation learning (Ross et al., 2011). The approach is compared against a pipeline approach with both Stanford NERC and FIGER for the NEC component and a subsequent distantly supervised RE. The assumption is that a joint approach with imitation learning outperforms a distant supervision approach with supervised NERC as a preprocessing step for some of the relations. Those relations are the ones between "non-standard" NE types such as "album", which do not correspond directly to a NE type the supervised NERC is trained for.

3.5 Training and Feature Extraction

3.5.1 Training

Distant supervision approaches use a variety of different learning methods, ranging from simple classifiers such as SVMs or MaxEnt models to tensor models or RNNs. For experiments on selecting training samples (Section 3.3), a simple MaxEnt classifier is used for the purpose of comparing

against Mintz et al. (2009) as a baseline.

For experiments on joint named entity recognition and relation extraction, the imitation learning algorithm DAgger (Ross et al., 2011) is used. The aim is to study if the same distantly labelled training data can successfully be used to train two models, a named entity classifier and a relation classifier, to outperform an approach which uses a supervised NEC and a distantly supervised RE, as described in Section 3.4.

3.5.2 Feature Extraction

Most distant supervision approaches use standard relation extraction features, such as the context around the relation candidate, the words between the subject and object candidate and the dependency path between the subject and object candidate (Mintz et al., 2009; Hoffmann et al., 2011).

For the experiments in Chapter 4 the relation features proposed in Mintz et al. (2009) are used for comparison reasons. In Chapter 6, feature selection is then studied in more detail. In particular, the goal is to study if low-precision high-frequency features such as bag of words features or high-precision low-frequency features such as lexicalised dependency paths, or a mix of those lead to the highest performance. Results reported in Mintz et al. (2009) suggest that there is very little difference between the performance of shallow features such as bag of words features and semantic features such as dependency features. However, for a multi-stage learning approach with NEC followed by RE, it is plausible that results could be different. The second stage (RE) is only reached if the first stage (NEC) indicates that the NEs are of the correct types. Therefore, it might be beneficial for the first stage to have high recall to make sure relevant RE candidates are not discarded. For the second stage, it might then be important to have high precision to make the correct prediction.

Another research goal is to study whether Web features can help NERC performance for RE with imitation learning. Although Web pages have been used for information extraction, this has so far not been studied. Using Web features is typically limited to information extraction from semi-structured data such lists and tables (Dalvi et al., 2012; Wang et al., 2012b; Shen et al., 2012) or to research on using Wikipedia as a corpus for named entity linking (Bunescu and Pasça, 2006; Han et al., 2011). Those studies indicate that HTML markup on Web pages helps to improve performance of semi-structured information extraction. In the case of named entity linking, Web pages with links are useful because they provide a corpus annotated with references to a knowledge base, which can then be used for learning to link named entities in text to a knowledge base.

As mentioned before (Section 3.2.2), Wikipedia is a curated corpus, and conclusions reached on the basis of studies of information extraction from Wikipedia might not hold for information extraction from Web pages in general. Specifically, in Wikipedia, links in articles almost always point to other articles, which are in turn often NEs. On general Web pages, many links are links to other websites and this assumption cannot be made.

The research goal is to study if features extracted from HTML markup such as links, text in bold or italics, or also lists can help improve NERC performance. Both local (the same mention) and global (on the same Web page) features are studied.

3.6 Selecting Testing Instances and Combining Predictions

3.6.1 Selecting Testing Instances

Testing instances for distant supervision are usually generated in the same way as training instances, holding some of the distantly annotated data out for testing. However, only very few sentences contain both the name of the subject and the object of a relation, i.e. some might be referred to using a personal pronoun or a definite description ("the artist"). While this is not a problem for training – more training data can be generated easily – by only using those sentences with names as mentions, some types of relations might be strategically missed. This is especially true for using Wikipedia articles or other websites containing person descriptions. The first sentence often contains the name, birthdate and birthplace of a person, thus containing both the subject and the object for the "birthdate" and "birthplace" relations. Other relations less central to a person's identity, e.g. for musical artists the names of their albums, are mentioned in other parts of the text and are less likely to contain a mention of the subject's name.

Therefore, another task that is important for knowledge base population is co-reference resolution. The aim is to evaluate performance using both standard co-reference resolution approaches and other simple heuristics based on gazetteers, e.g. to take into account that NEs are often referenced using a definite description ("the artist"). In Chapter 4, such co-reference methods are tested to extract relations from sentences which do not contain the name of the subject of the relation directly. The effect of additional predictions for those testing instances on knowledge base population is evaluated.

3.6.2 Combining Predictions

For instance-level relation extraction, predictions with the same surface form are combined for knowledge base population. Most studies on distant supervision combine those in a straightforward way after extraction (e.g. Hoffmann et al. (2011)). However, there are other methods for combining predictions, e.g. training an ensemble classifier to combine predictions (Viswanathan et al., 2015) or coupled learning (Carlson et al., 2010b). There is even a shared task, TAC KBP Slot Filler Validation[4].

Another simple way of combining extractions is to combine feature vectors of testing instances for the same $< s, o >$ tuples before training (Mintz et al., 2009). Chapter 4 contains experiments for testing which method for combining predictions is better – trying to combine feature vectors of the same relation tuples before training or combining the output of the classifiers.

Further, distantly supervised relation extraction approaches typically do not make use of background knowledge in the knowledge base for assessing which predictions to return. This includes how many objects there are for each subject and relation, e.g. how many albums are typically listed for each musical artist in the knowledge base. Such information could be used to assess how many results per subject and relation to return. Further, cross-relation information can be

[4] http://www.nist.gov/tac/2015/KBP/SFValidation/index.html

retrieved such as if objects which are related to the same subject have relations with the same object lexicalisations. An example for this would be the origin of a river, which is also a location contained by the same river. Chapter 4 further contains experiments on how to utilise such information from the knowledge base.

3.7 Summary

This chapter explains how different contributions of this PhD thesis fit together and can be integrated in a distant supervision framework. The following three chapters now detail methods and experiments conducted and Chapter 7 draws conclusions and discusses current and future work on the topics covered in this thesis.

Chapter 4

Distant Supervision for Web Relation Extraction

As discussed in Section 2.3.3, distant supervision techniques are very promising as they can be used to train relation extractors with respect to a relation schema, but without the need for manually labelled training data. However, they have several limitations with respect to Web information extraction that require further research. This chapter makes contributions with respect to six different aspects of distant supervision[1].

Selecting Training Instances: The distant supervision heuristic for automatically annotating training data can lead to false positives. Previous work has approached this shortcoming by adding training data or improving training data over time with active learning, changing how relations are modelled using the "at-least-one" assumption or by preprocessing the training data with unsupervised learning methods that try to distinguish suitable from unsuitable contexts (Section 2.4.3). Most of those approaches either require more direct supervision or are computationally expensive. The research documented in this chapter proposes a computationally inexpensive approach based on the notion of assessing how likely it is for lexicalisations of objects to be ambiguous. The assumption is that the more likely it is for an object candidate lexicalisation to be ambiguous, the more likely that candidate is to be a false positive. Ambiguity is measured using simple statistical methods based on data already present in the knowledge base. If a training instance is considered to be too ambiguous, it is considered unreliable and discarded from the training set. The benefit of this approach compared with other approaches is that it does not result in an increase of run-time during testing and is thus more suited towards Web-scale extraction than approaches which aim at resolving ambiguity during both training and testing. Moreover, since the approach assesses the ambiguity of an object candidate lexicalisation alone, it could be combined with other inexpensive existing approaches which distinguish suitable from unsuitable contexts (Section 2.4.3). Results show that the simple statistical methods proposed in

[1]This content in this chapter is based on publications at the third workshop on Semantic Web and Information Extraction at the 25th International Conference on Computational Linguistics (Augenstein, 2014b), in the proceedings of the 19th International Conference on Knowledge Engineering and Knowledge Management (Augenstein et al., 2014), and in the Semantic Web Journal (Augenstein et al., 2016a).

this chapter can increase the precision of distant supervision by filtering ambiguous training data.

Whereas this chapter focusses on discarding false positive training samples, this is related to the broader challenge of selecting suitable training instances is for machine learning problems (Blum and Langley, 1997). The goal of *representative sampling* is to select training samples which are representative of the learning problem, i.e. some training instances better aid the learning process than others. The main reason for studying selecting representative training instances is to achieve better generalisation by training on data which is more representative of the learning problem. Training on a non-representative sample, on the other hand, leads to learning a biased model which is unlikely to perform well on the testing dataset. Further reasons are computational efficiency, increasing the speed of learning and, related to the problem discussed in this thesis, the cost of annotation. Samples are either selected before training or during training, which is known as *learning to sample* (Bouchard et al., 2015).

Arctic Monkeys are a guitar rock band from Sheffield, England. **The group**, which is comprised of frontman and lyricist Alex Turner, guitarist Jamie Cook, drummer Matt Helders and bassist Nick O'Malley, are one of the most successful British bands of the 21st century: their debut album 'Whatever People Say I Am, That's What I'm Not' is the fastest-selling debut in British chart history and they have released five consecutive Number One albums. **They** have released two albums, 'Whatever...' and their most recent LP 'AM', which have received 10/10 reviews from NME. Other accolades and achievements include winning seven Brit Awards and headlining Glastonbury Festival on two occasions.

Figure 4.1: Arctic Monkeys biography, illustrating discourse entities[2]

Selecting Testing Instances: Existing distant supervision systems only learn to extract relations from sentences which contain an explicit mention of the name of both the subject and the object of a relation (see Section 2.4.2). As a result, those sentences which contain an indirect reference to the subject, e.g. using a pronoun or a category, are not considered for testing. Further, some information is systematically missed out on due to the nature of discourse. Figure 4.1 illustrates this with a typical musical artist biography, as often found on Web pages: the first sentence contains a direct mention of the name of the musical artist ("Arctic Monkeys") and relations representing key facts about the artist ("guitar rock" genre, their origin is "Sheffield, England"). The second sentence refers to "Artic Monkeys" as "the group" and mentions the band members and their debut album, while the third sentence utilises the pronoun "they" as a referent and lists two more albums. The fourth sentence finally does not contain any referent for "Artic Monkeys". If only sentences containing direct mentions of "Arctic Monkeys" are used for testing, other sentences containing mentions in the form of pronouns ("they"), categories ("the group") or no explicit referents are missed out on. The contribution of this chapter to selecting testing instances is therefore to propose and evaluate methods to extract relations from sentences which do not contain direct mentions. One method is to use those sentences contained in a paragraph which contain at least one sentence with a direct mention. Another method is the integration of an

[2] http://www.nme.com/artists/arctic-monkeys

existing supervised co-reference resolution approach. This does not always perform well, because the co-reference resolution approach might be trained for a different domain or genre. It also partly relies on other NLP components, e.g. the recognition of NEs, which might also be trained for a different domain or genre and not perform well, resulting in errors being propagated to the co-reference resolution stage. A third method therefore is to perform co-reference resolution, but in a less restrictive way, by only applying gender and number gazetteers for co-reference resolution. The evaluation shows that a combination of the supervised co-reference resolution approach and gazetteers performs best.

Co-reference resolution for distant supervision has been tested by one related work (Koch et al., 2014), which was published at the same time as the publications this chapter is based on (Augenstein, 2014b; Augenstein et al., 2014, 2016a). However, they assume the text already contains co-reference annotations and evaluate on the ACE 2005 corpus (Doddington et al., 2004). This is not portable to other corpora without manual annotation and therefore a much less realistic scenario than the one described in this thesis.

Recognising Entities: Distant supervision approaches typically use named entity recognisers and classifiers trained for either the news genre or Wikipedia (Section 2.4.4). These typically label entities as either persons (PER), locations (LOC), organisations (ORG), subtypes thereof, or miscellaneous (MISC). The definition of what constitutes the MISC category differs widely from corpus to corpus, which makes it the most difficult NE type to recognise, and specifically to adapt to other genre. When applying such approaches to heterogenous Web pages, types of entities which would fall into the MISC category specifically, but also subclasses of person, location and organisation are often not recognised. Two of those types used for the experiments described in this thesis are *MusicalArtist:track* and *MusicalArtist:album*. NERCs often struggle with long NE mentions, such as "Whatever People Say I Am, That's What I'm Not" (Figure 4.1). The proposed approach in this chapter is to use additional heuristics based on HTML markup on Web pages to recognise such mentions.

Setting: As described in Section 3.2.1, existing distant supervision approaches generally assume that every text might contain information about any possible relation. This means, when identifying candidates for specific relations, they search the whole corpus for every relation pair from a background knowledge base, which is computationally expensive. In addition, the trained classifiers (or classifier, for multi-class settings) also have to learn to distinguish between all possible relations, which is infeasible with diverse domains and a big corpus.
Attempting to match each named entity in the corpus with each named entity in a background knowledge base is a big computational effort, which should be avoided, if possible. In reality, Web pages are often about a specific entity (see Figure 4.1 for an example). What would reduce this effort would be to determine what entities specific Web pages are about and then use only those Web pages to generate training data for those entities. This could also increase the precision of training data, especially for negative training.

Existing approaches sample negative training and testing data from sentences in any document which contain two NEs which are unrelated according to the knowledge base. Sampling negative and positive training data from the same distribution is generally a good strategy for achieving

high model performance (Li et al., 2010). We use a different strategy for selecting testing data from existing distant supervision work. We select negative testing data from Web pages not from all sentences in the corpus, but only from Web pages which are assumed to be about a specific entity using an information retrieval step. Therefore, we select negative training data in the same way.

The contribution of this chapter with regard to the setting is therefore the proposal of an architecture for a Web search-based distant supervision approach which gives instance-level results for knowledge base population. The search-based aspect of the setting mimics a user with a specific query, e.g. "What albums did the musical artist Arctic Monkeys release?" For search, a commercial search engine is used with the assumption that top ranked retrieved Web searches are mostly relevant to the query.

Evaluation and Corpus: Since no publicly available corpus for Web-based distant supervision exists (see Section 2.4.1) a corpus is collected using the setting introduced above. The knowledge base Freebase (Bollacker et al., 2008) is used, in line with most existing work, from which the most popular types and relations are selected for corpus creation. A hold-out evaluation is then performed on sentence-level and on instance-level.

Combining Predictions: For instance-level relation extraction, predictions are combined for knowledge base population. Most previous approaches combine predictions in a straightforward way after extraction (see Section 2.4.2). Another simple way of combining extractions is to combine feature vectors of testing instances for the same $< s, o >$ tuples before training, which is how experiments in Mintz et al. (2009) are performed. However, the two different methods are not compared. This chapter therefore contains experiments for testing what a more successful method for combining predictions is – trying to combine feature vectors of the same relation tuples before training or combining the output of the classifiers. Further, distantly supervised relation extraction approaches typically do not make use of background knowledge in the knowledge base for assessing which predictions to return. This includes how many objects there are for each subject and relation, e.g. how many albums are typically listed for each musical artist in the knowledge base. Also cross-relation information can be retrieved such as if objects which are related to the same subject have relations with the same object lexicalisations. This chapter further contains experiments on how to utilise such information from the knowledge base for assessing which predictions to return.

4.1 Distantly Supervised Relation Extraction

Distantly supervised relation extraction is defined as automatically labelling a corpus with properties, P and resources, R, where resources stand for entities E from a knowledge base, KB, to train a classifier to learn to predict binary relations. The distant supervision paradigm is defined as follows (Mintz et al., 2009):

> If two entities participate in a relation, any sentence that contains those two entities might express that relation.

In general relations are of the form $(s, p, o) \in E \times P \times E$, consisting of a subject, a predicate and an object; during training, those which are contained in a knowledge base are considered, i.e. $(s, p, o) \in KB \subset R \times P \times R$. In any single extraction we consider only those subjects in a particular class $C \subset R$, i.e. $(s, p, o) \in KB \cap C \times P \times R$. Each resource $r \in R$ has a set of lexicalisations, $L_r \subset L$. Lexicalisations are retrieved from the KB, where they are represented as the name or alias, i.e. less frequent name of a resource.

In the remainder of this chapter, several variations of this approach are presented, method names are indicated in bold font.

4.2 Training Data Selection

Before using the automatically labelled corpus to train a classifier, training examples containing highly ambiguous lexicalisations are detected and discarded. What is measured is the degree to which a lexicalisation $l \in L_o$ of an object o is ambiguous. Ambiguity is defined as the number of senses the lexicalisation has, where the number of senses is the number of unique resources representing a lexicalisation.

4.2.1 Ambiguity Of Objects

A first approach is to discard lexicalisations of objects if they are ambiguous for the subject entity, i.e. if a subject is related to two different objects which have the same lexicalisation, and express two different relations. To illustrate this, consider the following problem: *Let It Be* can be both an *album* and a *track* of the subject entity *The Beatles*, therefore *Let It Be* should be discarded as a training example for the class *Musical Artist*.

Unam: For a given subject s, if it is discovered that a lexicalisation for a related entity o, i.e. $(s, p, o) \in KB$ and $l \in L_o$, then it may be the case that $l \in L_r$ for some $R \ni r \neq o$, where also $(s, q, r) \in KB$ for some $q \in P$, i.e. l has a "sense" o and r, giving rise to ambiguity. Next, A_l^s, is defined as the ambiguity of a lexicalisation with respect to the subject as follows: $A_l^s = |\{r \mid l \in L_o \cap L_r \wedge (s, p, o) \in KB \wedge (s, q, r) \in KB \wedge r \neq o\}|$.

4.2.2 Ambiguity Across Classes

In addition to being ambiguous for a subject of a specific class, lexicalisations of objects can be ambiguous across classes. The assumption made here is that the more senses an object lexicalisation has, the more likely it is that the object occurrence is confused with an object lexicalisation of a different property of any class. An example for this are common names of book authors or common genres as in the sentence "*Jack* mentioned that he read *On the Road*", in which *Jack* is falsely recognised as the author Jack Kerouac.

Stop: One type of very ambiguous words with many senses is stop words. Since some objects of relations in the training set might have lexicalisations which are stop words, those lexicalisations are discarded if they appear in a stop word list. For this purpose, the stop word list described

in Lewis et al. (2004) is used, which was originally created for the purpose of information retrieval and contains 571 highly frequent words.

Stat: For other highly ambiguous lexicalisations of object entities the approach is to estimate cross-class ambiguity, i.e. to estimate how ambiguous a lexicalisation of an object is compared with other lexicalisations of objects of the same relation. If its ambiguity is comparatively low, it is considered a reliable training instance, otherwise it is discarded. For the set of classes under consideration, the set of properties that apply are known, $D \subset P$, and the sets $\{o \mid (s,p,o) \in KB \wedge p \in D\}$ can be retrieved, as well as the set of lexicalisations for each member, L_o. A_o is then computed, which is the number of senses for every lexicalisation of an object L_o, where $A_o = |\{o \mid l \in L_o\}|$.

The number of senses of each lexicalisation of an object per relation is viewed as a frequency distribution. Several metrics are computed – min, max, median ($Q2$), the lower ($Q1$) and the upper quartile ($Q3$) of those frequency distributions – and compared to the number of senses of each lexicalisation of an object. If $A_l > Q$, where Q is either $Q1$, $Q2$ or $Q3$ depending on the model, the lexicalisation of the object is discarded.

StatRes: Since Stat is mainly aimed at n-ary relations, for which many training instances are available, the goal of StatRes is to restrict the impact of Stat for relations with only few object lexicalisations per relation. The number of object lexicalisations per property is computed and viewed as a frequency distribution with min, max, median, lower and upper quartile. If the number of object lexicalisations at the upper quartile for a relation is 2 or smaller, no training instances for that relation are discarded. This method is applied for all variants of StatRes.

4.2.3 Relaxed Setting

In addition to increasing the precision of distantly supervised systems by filtering training data, further experiments are performed aimed at increasing recall by changing the method for creating test data. Instead of testing, for every sentence, if the sentence contains a lexicalisation of the subject and one additional entity, we relax the former restriction. The assumption made here is that the subject of the sentence is mostly consistent within one paragraph as the use of paragraphs usually implies a unit of meaning, i.e. that sentences in one paragraph often have the same subject. In practice this means that classifiers are first trained using the original assumption and then, for testing, instead of only extracting information from sentences which contain a lexicalisation of the subject, information is also extracted from sentences which are in the same paragraph as a sentence which contains a lexicalisation of the subject. The new relaxed distant supervision assumption is as follows:

> If two entities participate in a relation, any *paragraph* that contains those two entities might express that relation, even if not in the same sentence, provided that another sentence in the paragraph in itself contains a relationship for the same subject of the relation.

If the assumption is relaxed so two entities only have to appear together in a paragraph, that

means that the subject has to be resolved in a different way, e.g. by performing co-reference resolution and searching for a pronoun which is coreferent with the subject mention in a different sentence. Four different methods are tested for the relaxed setting, one of which does not attempt to resolve the subject of sentences, one based on an existing co-reference resolution tool, and two based on gazetteers of Web co-occurrence counts for number and gender of noun phrases.

NoSub: Instead of trying to perform co-reference resolution, the first approach does not attempt to find the subject of the sentence at all. Instead, all features which require the position of the subject of the relation to be known are disregarded. Features used in both the NoSub setting and the normal setting are documented in Section 4.3.4.

CorefS: To test how useful off-the-shelf co-reference resolution is for a variety of different classes and properties, co-reference resolution using the Stanford NLP co-reference resolution tool is performed. For every sentence in a paragraph that contains at least one sentence with the subject entity, if any of the sentences contain a pronoun or noun phrase that is coreferent with the subject entity, it is treated as if it were a lexicalisation of the subject entity and all features are extracted which are also extracted for the normal setting.

CorefN and **CorefP**: Since the Stanford NLP co-reference resolution tool is a supervised approach trained on the news genre, it might not be able to resolve co-references for some of the classes used. Since training data is not available for all of the domains considered, a heuristic based on Web co-occurrence counts using the gazetteers collected by Bergsma and Lin (2006) is used instead.

The first step in co-reference resolution is usually to group all mentions in a text, i.e. all noun phrases and pronouns, by gender and number. If two mentions disagree in number or gender, they cannot be coreferent. As an example, should "The Beatles" and "he" be found in the same sentence, then "The Beatles" and "he" could not be coreferent, because "The Beatles" is a plural neutral noun phrase, whereas "he" is a singular male pronoun. However, a-priori information on number and gender of the subject entity is not available. Therefore, those judgments are instead made based on the number and gender of the class of the subject, e.g. The Beatles is a Musical Artist, which can be a band (plural) or a female singer or a male singer. Bergsma and Lin (2006) have collected such a resource automatically, which also includes statistics to assess how likely it is for a noun phrase to be a certain number or gender. In particular, they collected co-occurrence counts of different noun phrases with *male*, *female*, *neutral* and *plural* pronouns using Web search. The heuristic co-reference approach consists of three steps. First, noun phrases which express general concepts related to the subject entity are collected, which is here referred to as *synonym gazetteer*. The process starts with the lexicalisation of the class of the entity (e.g. "Book"), for which synonyms, hypernyms and hyponyms are retrieved using Wikipedia redirection pages and WordNet (Fellbaum, 1998). Second, the gender of each class is determined by looking up co-occurrence counts for each general concept in the noun phrase, gender and number gazetteer. Co-reference counts are aggregated for each class and gender or number (i.e. male, female, neutral, plural). If the aggregated count for each number or gender is at least 10% of the total count for all genders and numbers, that gender or number is considered to *agree with* the class. For each class, a *pronoun gazetteer* is then created, which contains all male,

female, neutral or plural personal prounouns including possessives, e.g. for "Book", that gazetteer would contain "it, its, itself". Lastly, those gazetteers are used to resolve co-reference. For every sentence in a paragraph that contains at least one sentence with the subject entity, if any of the following sentences contain a pronoun or noun phrase that is part of the synonym or pronoun gazetteer for that class and it appears in the sentence before the object lexicalisation, that noun phrase or pronoun is considered coreferent with the subject. The reason to only consider noun phrases or pronouns to be coreferent with the subject entity if they appear after the object entity is to improve precision, since anaphora (expressions referring back to the subject) are far more common than cataphora (expressions referring to the subject appearing later in the sentence).

Two different methods are tested. CorefN only uses the synonym gazetteer, whereas CorefP uses both the synonym and the pronoun gazetteer. If a sentence contains both a synonym and a pronoun, the synonym is selected as coreferent for the subject. Then, as for CorefS, those noun phrases and pronouns are treated as lexicalisations of the subject and all features also used for the normal setting are extracted.

4.2.4 Information Integration

After features are extracted, a classifier is trained and used to make predictions for testing instances. Predictions for different testing instances can be combined for populating knowledge bases. Since the same relations might be found in different documents, but some contexts might be inconclusive or ambiguous, it is useful to integrate information taken from multiple predictions to increase the chances of predicting the correct relation. Several different methods are tested to achieve this.

Comb: Normally, one feature vector would be extracted for each training or testing instance. However, the context of individual instances might be sparse and inconclusive. Since training and testing instances often occur several times in the document, features can be extracted from the individual instances, and then aggregated to one feature vector. This provides the classifier with more information per training or testing instance. This is the default way of integrating information used in Mintz et al. (2009).

Aggr: For every Freebase class, all testing instances are retrieved from the corpus and the classifier's confidence values for classes assigned to object occurrences. There are usually several different predictions, e.g. the same occurrence could be predicted to be MusicalArtist:album, MusicalArtist:origin and MusicalArtist:NONE. For a given lexicalisation l, representing an object to which the subject is related, the classifier gives each object occurrence a prediction which is the combination of a predicted relation and a confidence. These are collected across the chosen documents to form a set of confidence values, for each predicted relation, per lexicalisation E_p^l. For instance if the lexicalisation l occurs three times across the documents and is predicted to represent an object to relation p_1 once with confidence 0.2, and in other cases to represent the object to relation p_2 with confidence 0.1 and 0.5 respectively, then $E_{p_1}^l = 0.2$ and $E_{p_2}^l = \{0.1, 0.5\}$. Following this, only the relation p with the highest single confidence value $E > 0.5$ is selected. In order to form an aggregated confidence for each relation with respect to the lexicalisation, g_l^p, the mean average for each such set is calculated and normalised across relations, as follows:

$g_p^l = \overline{E_p^l} \cdot \frac{|E_p^l|}{\sum_{q \in P} |E_q^l|}$. For each lexicalisation l, the relation p with the highest confidence g_p^l is selected.

Limit: One of the shortcomings of Aggr is that it returns all possible aggregated predictions for each relation, which sometimes means too many predictions are returned. To address this, the number of object lexicalisations per property is computed and viewed as a frequency distribution. Maximum and upper quartile of that distribution is computed, then all predictions are sorted by confidence value in descending order. The highest ranked n predictions are selected and returned, starting with the one with the highest confidence value. For LimitMax n is the maximum of the object lexicalisation per property frequency distribution, whereas for Limit75 it is the upper quartile.

Multilab: Another shortcoming of Aggr is that it can only be used to predict one label per aggregated prediction, i.e. *Let it Be* will either be predicted to be MusicalArtist:album or MusicalArtist:track, but not both. While it is possible to train a multi-label classifier with noisy, ambiguous examples (Surdeanu et al., 2012), another option, which is pursued here, is to discard those examples for training, and to integrate them for testing post hoc. To find out which relations have any object lexicalisations overlapping with other relations, this information about *mutual labels* is collected from the part of Freebase used for training. After predictions are aggregated using Aggr, instead of only returning the label with highest confidence, all possible labels are sorted by confidence value. If the label with highest confidence and the one with second highest confidence are mutual labels, both of them are returned, afterwards, if the label with highest confidence and the one with third highest confidence are mutual labels, the label with third highest confidence is also returned.[3]

4.3 System

4.3.1 Corpus

To create a corpus for Web relation extraction using background knowledge from Linked Data, seven Freebase classes and their five to seven most prominent properties are selected, as shown in Table 4.1. The selected classes are subclasses of either "Person" (Musical Artist, Politician), "Location" (River), "Organisation" (Business (Operation)), Education(al Institution)) or "Mixed" (Film, Book). To avoid noisy training data, only entities which have values for all of those properties are used and retrieved with the Freebase API. This resulted in 1800 to 2200 entities per class. For each entity, at most 10 Web pages were retrieved via the Google Search API using the search pattern "'*subject_entity*" *class_name relation_name*', e.g. "'The Beatles" Musical Artist Origin'. By adding the class name, the expectation is for the retrieved Web pages to be more relevant to the extraction task. Though adding the class name means the Web pages are more relevant, this might boost the results compared to other distant supervision methods which do not make use of this information. Although subject entities can have multiple lexicalisations, Freebase distinguishes between the most prominent lexicalisation (the entity name) and other lexicalisations

[3]There is only one instance of three mutual labels for the evaluation set, namely *River:origin*, *River:countries* and *River:contained by*

(entity aliases). The entity name is used for all of the search patterns. In total, the corpus consists of around one million pages drawn from 76,000 different websites. An overview of the distribution of websites per class is given in Table 4.2.

Person	
Musical Artist : album	Politician : birthdate
Musical Artist : active (start)	Politician : birthplace
Musical Artist : active (end)	Politician : educational institution
Musical Artist : genre	Politician : nationality
Musical Artist : record label	Politician : party
Musical Artist : origin	Politician : religion
Musical Artist : track	Politician : spouses
Organisation	
Business : industry	Education : school type
Business : employees	Education : mascot
Business : city	Education : colors
Business : country	Education : city
Business : date founded	Education : country
Business : founders	Education : date founded
Mixed	
Film : release date	Book : author
Film : director	Book : characters
Film : producer	Book : publication date
Film : language	Book:genre
Film : genre	Book : original language
Film : actor	
Film : character	
Location	
River : origin	
River : mouth	
River : length	
River : basin countries	
River : contained by	

Table 4.1: Freebase classes and properties/relations used

4.3.2 NLP Pipeline

Text content is extracted from HTML pages using the Jsoup API,[4] which strips text from each element recursively. Each paragraph is then processed with Stanford CoreNLP[5] to split the text into sentences, tokenise it, annotate it with part of speech (POS) tags and normalise time expressions. Named entities are classified using the 7 class (time, location, organisation, person, money, percent, date) named entity model. For the relaxed setting (Section 4.2.3), co-references

[4] http://jsoup.org
[5] http://nlp.stanford.edu/software/corenlp.shtml

Musical Artist		Politician	
21	en.wikipedia.org	17	en.wikipedia.org
6	itunes.apple.com	4	www.huffingtonpost.com
5	www.allmusic.com	3	votesmart.org
4	www.last.fm	3	www.washingtonpost.com
3	www.amazon.com	2	www.nndb.com
2	www.debate.org	2	www.evi.com
2	www.reverbnation.com	2	www.answers.com
57	Others	67	Others
Business		**Education**	
13	en.wikipedia.org	23	en.wikipedia.org
6	www.linkedin.com	8	www.linkedin.com
2	www.indeed.com	4	colleges.usnews.rankingsandreviews.com
2	www.glassdoor.co.uk	1	www.forbes.com
1	connect.data.com	1	www.facebook.com
1	www.answers.com	1	www.greatschools.org
1	www.forbes.com	1	www.trulia.com
74	Others	61	Others
Film		**Book**	
15	en.wikipedia.org	20	en.wikipedia.org
15	www.imdb.com	15	www.goodreads.com
3	www.amazon.com	12	www.amazon.com
3	www.rottentomatoes.com	9	www.amazon.co.uk
1	www.amazon.co.uk	4	www.barnesandnoble.com
1	www.tcm.com	3	www.abebooks.co.uk
1	www.nytimes.com	2	www.abebooks.com
61	Others	28	Others
River			
24	en.wikipedia.org		
2	www.britannica.com		
1	www.researchgate.net		
1	www.facebook.com		
1	www.gaiagps.com		
1	www.tripadvisor.co.uk		
1	www.encyclo.co.uk		
69	Other		

Table 4.2: Distribution of websites per class in the Web corpus sorted by frequency

are resolved using the Stanford CoreNLP co-reference resolution system (Lee et al., 2013)[6].

Relation candidate identification

Some of the relations we want to extract values for cannot be categorised according to the 7 classes detected by the Stanford NERC and are therefore not recognised. An example for this is *MusicalArtist:album*, *MusicalArtist:track* or *MusicalArtist: genre*. Therefore, as well as recognising named entities with Stanford NERC as relation candidates, NER heuristics are applied, which only recognise entity boundaries, but do not classify them. Note that this does not solve the problem of labelling NEs for which Stanford NERC fails – a novel method for solving this is proposed in Chapter 6. Since the NE type is used as a feature for relation extraction (see Section 4.3.4), if an NE is recognised with heuristics, the NE feature is set to "O" (no type), which is the same as for NEs for which Stanford NER cannot detect a NER type.

To detect entity boundaries, sequences of nouns and sequences of capitalised words are recognised and both greedy and non-greedy matching is applied, i.e. both full sequences and subsequences of those sequences are considered. The reason to do greedy as well as non-greedy matching is because the lexicalisation of an object does not always span a whole noun phrase, e.g. while 'science fiction' is a lexicalisation of an object of *Book:genre*, 'science fiction book' is not. However, for *MusicalArtist:genre*, 'pop music' would be a valid lexicalisation of an object. For greedy matching, whole noun phrases and sequences of capitalised words are considered. For non-greedy matching, we consider all subsequences starting with the first word of the those phrases as well as single tokens, i.e. for 'science fiction book', the candidates considered are 'science fiction book', 'science fiction', 'science', 'fiction' and 'book'. Further, short sequences of words in quotes are recognised. This is because lexicalisation of objects of *MusicalArtist:track* and *MusicalArtist:album* often appear in quotes, but are not necessarily noun phrases.

4.3.3 Annotating Sentences

The next step is to identify which sentences express relations of interest. Only sentences from Web pages which were retrieved using a query which contains the subject of the relation are used. To annotate sentences, all lexicalisations L_s, L_o for subjects and objects related under properties P for the subject's class C are retrieved from Freebase. Next, it is checked, for each sentence, if the sentence contains at least two entities recognised using either the Stanford NERC or the NE heuristics described in Section 4.3.2, one of which has a lexicalisation of a subject and the other a lexicalisation of an object of a relation. If it does, this sentence is used as training data for that property. All sentences which contain a subject lexicalisation and one other entity that is not a lexicalisation of an object of any property of that subject are used as negative training data for the classifier. It is likely that some of those instances selected as negative training data are false negatives since knowledge bases are often incomplete (Min et al., 2013). Mintz et al. (2009) only use 1% of their negative training data; however, in the setting described in this chapter, all training data is used. This is because there is less training data overall and it was observed during the

[6] http://nlp.stanford.edu/software/dcoref.shtml

evaluation that using more negative training data increases precision and recall of the system. For testing all sentences that contain at least two entities recognised by either entity recogniser, one of which must be a lexicalisation of the subject, are used. For the relaxed setting (Section 4.2.3) only the paragraph the sentence is in must contain a lexicalisation of the subject.

4.3.4 Training Data Selection

After training data is retrieved by automatically annotating sentences, instances are selected from it, or rather some of the training data is discarded, according to the different methods outlined in Section 4.2. These models are compared against Baseline models, which do not discard any training instances.

Features

Given a relation candidate as described in Section 4.3.2, the system then extracts the following lexical features and named entity features, some of them also used by Mintz et al. (2009). Features marked with (*) are only used in the normal setting, but not for the NoSub setting(Section 4.2.3).

- The object occurrence
- The bag of words of the occurrence
- The number of words of the occurrence
- The named entity class of the occurrence assigned by the 7-class Stanford NERC
- A flag indicating if the object or the subject entity came first in the sentence (*)
- The sequence of POS tags of the words between the subject and the occurrence (*)
- The bag of words between the subject and the occurrence (*)
- The pattern of words between the subject entity and the occurrence (all words except for nouns, verbs, adjectives and adverbs are replaced with their POS tag, nouns are replaced with their named entity class if a named entity class is available) (*)
- Any nouns, verbs, adjectives, adverbs or named entities in a 3-word window to the left of the occurrence
- Any nouns, verbs, adjectives, adverbs or named entities in a 3-word window to the right of the occurrence

In comparison with Mintz et al. (2009) a richer feature set is used, specifically more bag of words features, patterns, a numerical feature and a different, more fine-grained named entity classifier.
Experiments are performed for both relation extraction for knowledge base population, as in Mintz et al. (2009), and for predicting relations for individual testing instances. For knowledge base population, feature vectors are aggregated for relation tuples, i.e. for tuples with the same subject and object, for training a classifier. In contrast, predicting relations for individual testing instances means that feature vectors are not aggregated for relation tuples. While predicting relations is sufficient if the goal is only to retrieve a list of values for a certain property, and not to annotate text with relations, combining feature vectors for distant supervision approaches can introduce additional noise for ambiguous subject and object occurrences.

4.3.5 Models

The models evaluated differ with respect to how sentences are annotated for training, how positive training data is selected, how negative training data is selected, which features are used, how sentences are selected for testing and how information is integrated.

Mintz: This model follows the setting of the model which only uses lexical features described in Mintz et al. (2009). Sentences are annotated using the Stanford NERC (Finkel et al., 2005) to recognise subjects and objects of relations, 1% of unrelated entities are used as negative training data and a basic set of lexical features is used. If the same relation tuple is found in several sentences, feature vectors extracted for those tuples are aggregated. For testing, all sentences containing two entities recognised by the Stanford NERC are used.

Baseline: This group of models follows the setting described in Section 4.3. It uses sentences annotated with both Stanford NERC and NER heuristics (Section 4.3.2). All negative training data is used. For testing, all sentences containing two entities recognised by both Stanford NERC and the NER heuristics are used.

Comb: This group of models uses the same settings as Baseline models except that feature vectors for the same relation tuples are aggregated.

Aggr, Limit, MultiLab: These models use the same strategy for named entity recognition and selecting negative training data as the Comb group of models. However, feature vectors are not aggregated. Instead, labels are predicted for testing instances and relations are predicted using the different information integration methods described in Section 4.2.4.

Unam, Stop, Stat, StatRes: Those models select training data according to the different strategies outlined in Section 4.2.

NoSub: This group of models uses the relaxed setting described in Section 4.2.3 which does not require sentences to explicitly contain subjects and only uses a restricted set of features for testing which do not require the position of the subject entity to be known.

CorefS: This is a variant of the relaxed setting, also described in in Section 4.2.3 which uses Stanford Coref to resolve co-references. The full set of features is extracted for testing.

CorefN, CorefP: Co-references are resolved for those variants of the relaxed setting using gender and number gazetteers. As for CorefS, the full set of features is extracted for testing.

4.3.6 Predicting Relations

In order to be able to compare the results, the same classifier as in Mintz et al. (2009) is chosen, a multi-class logistic regression classifier. One multi-class classifier per Freebase class and model is trained, i.e. 7 classifiers in total. The models are used to classify each testing instance into one of the relations of the class or NONE (no relation). Predictions are then aggregated for knowledge base population using the different information integration methods described in Section 4.2.4.

4.4 Evaluation

4.4.1 Manual Evaluation

To evaluate the models, first, a hold-out evaluation is carried out on a subset of Freebase types, for which 50% of the data per Freebase type is used for training and 50% for testing. The whole corpus is annotated with relations already present in Freebase, as described in Section 4.3. 50% of it is used for training and 50% for testing. Next, a manual evaluation of the highest ranked 10% of predictions is conducted for a subset of the classes. The following two metrics are used: number of predictions (number of occurrences which are predicted to be a value of one of the properties for an entity) and precision.

Ideally, recall would be reported, which is defined as the number of detected true positives divided by the number of positive instances. However, the number of positive instances is not known, those could only be obtained by manually examine the whole corpus.

The respective models are restricted as to how many positive predictions they can make by the distant supervision assumption or the relaxed distant supervision assumption. Therefore, instead of reporting recall, the number of true positives is reported. This equals the number of positive instances, also called *hits*, identified by manual labelling. For the manual evaluation, all predictions are ranked by probability per property and manually annotated and compared from the top ranked 10%, then are averaged over all properties per class, as shown in Table 4.3.

Model	**Book**		**Musical Artist**		**Film**		**Politician**	
	N	**P**	**N**	**P**	**N**	**P**	**N**	**P**
Mintz	105	0.236	216	0.255	110	0.343	103	0.241
Comb	168	0.739	510	0.672	283	0.764	150	0.863
Baseline	1546	0.855	2060	0.586	1574	0.766	488	0.868
Baseline + Stop + Unam	1539	0.857	2032	0.620	1574	0.766	485	**0.874**
Baseline + Stop + Unam + Stat75	1360	**0.948**	1148	**0.694**	303	**0.775**	474	0.82
Comb + NoSub	705	0.653	2363	0.619	973	0.623	363	0.687
Baseline + NoSub	4948	0.663	11286	0.547	2887	0.673	3970	0.703

Table 4.3: Manual evaluation results: Number of true positives (N) and precision (P) for all Freebase classes

4.4.2 Automatic Evaluation

The goal of the automatic evaluation is to measure how the different distant supervision models described in Section 4.3.5 perform for the task of knowledge base population, i.e. to measure how accurate the information extraction methods are at replicating the test part of the knowledge base.

The following metrics are computed: precision, recall and an estimated upper bound for recall. Precision is defined as the number of correctly labelled relations divided by the number of correctly labelled plus the number of incorrectly labelled relations. Recall is defined as the number of correctly labelled relations divided by the number of all relation tuples in the knowledge base. The number of all relation tuples includes all different lexicalisations of objects contained in the

knowledge base.

To achieve a perfect recall of 1, all relation tuples in the knowledge base have to be identified as relation candidates in the corpus first. However, not all relation tuples also have a textual representation in the corpus. To provide insight into how many of them do, we compute an estimated upper bound for recall. The upper bound would usually be computed by dividing the number of all relation tuples appearing in the corpus by the number of relation tuples in the knowledge base, as e.g. in Gentile et al. (2013). The upper bound provided is only an estimate, since the corpus is too big to examine each sentence manually. Instead, it is computed by dividing the number of relation tuples identified using the most inclusive relation candidate identification strategy, those used by the NoSub models, by the number of relation tuples in the test knowledge base.

Results for different training data selection models detailed in Section 4.3.5 averaged over all properties of each class are listed in Table 4.4. Results for different information integration models are listed in Table 4.5 and results for different co-reference resolution methods per class are listed in Table 4.6. Finally, Table 4.7 shows results for the best performing normal model and the best performing model for the relaxed setting per Freebase class.

Model	P	R	F1
Mintz	0.264	0.0359	0.0632
Baseline	0.770	**0.0401**	**0.0762**
Baseline + Stop + Unam	0.773	0.0395	0.0752
Baseline + Stop + Unam + Stat75	**0.801**	0.0243	0.0472
Baseline + Stop + Unam + Stat50	0.801	0.0171	0.0335
Baseline + Stop + Unam + Stat25	0.767	0.00128	0.0026
Baseline + Stop + Unam + StatRes75	0.784	0.0353	0.0676
Baseline + Stop + Unam + StatRes50	0.787	0.0341	0.0654
Baseline + Stop + Unam + StatRes25	0.78	0.0366	0.0699
NoSub	0.645	0.0536	0.0990
CorefS	0.834	0.0504	0.0951
CorefN	0.835	0.0492	0.0929
CorefP	0.830	**0.0509**	**0.0959**
CorefN + Stop + Unam + Stat75	**0.857**	0.0289	0.0559

Table 4.4: Training data selection results: micro average of precision (P), recall (R) and F1 measure (F1) over all relations, using the Multilab+Limit75 integration strategy and different training data selection models. The estimated upper bound for recall is 0.0917.

4.5 Results

From a manual evaluation of the highest ranked 10% of results per property (Section 4.3) it can be observed that there is a significant difference in terms of precision between the different model groups. In addition, it can be observed that there is a sizable difference in precision for different properties and classes. It is easiest to classify numerical values correctly, followed by people.

Model	P	R	F1
Comb	0.742	0.0328	0.0628
Aggr	0.813	**0.0341**	**0.0655**
LimitMax	0.827	0.0267	0.0517
MultiLab	0.837	0.0307	0.0336
Limit75 + MultiLab	**0.857**	0.0289	0.0432

Table 4.5: Information integration results: micro average of precision (P), recall (R) and F1 measure (F1) over all relations, using the CorefN+Stop+Unam+Stat75 model and different information integration methods.

Class	CorefS			CorefN			CorefP		
	P	R	F1	P	R	F1	P	R	F1
Musical Artist	0.736	0.0112	0.0221	0.744	0.0112	0.0221	**0.7473**	**0.01121**	**0.0221**
Politician	**0.796**	**0.0577**	**0.1076**	0.788	0.0498	0.0937	0.788	0.0567	0.1058
River	**0.890**	0.0902	0.1638	0.889	0.0902	0.1638	0.873	**0.0932**	**0.1684**
Business	0.849	0.1232	0.2152	**0.861**	0.1352	0.2337	0.856	**0.1593**	**0.2686**
Education	0.927	**0.09**	0.1641	**0.928**	0.0893	0.1629	0.926	0.0898	**0.1637**
Book	**0.814**	0.0465	0.0880	0.804	0.0461	0.0872	0.808	**0.0484**	**0.0913**
Film	0.8	0.0405	0.0771	**0.0803**	0.0411	0.0544	0.795	**0.0415**	**0.0789**

Table 4.6: Co-reference resolution results: micro average of precision (P), recall (R) and F1 measure (F1) over all relations, using the CorefN+Stop+Unam+Stat75 model and different co-reference resolution methods.

Class	best normal			best relaxed			
	P	R	F1	P	R	F1	upper bound
Musical Artist	0.671	0.006	0.1102	0.7443	0.0112	0.0354	0.0221
Politician	0.76	0.0316	0.0607	0.7876	0.0498	0.1777	0.0937
River	0.875	0.0234	0.0456	0.889	0.0902	0.14	0.1638
Business Operation	0.851	0.071	0.1311	0.8611	0.1352	**0.232**	0.2337
Educational Institution	**0.931**	**0.0795**	0.1465	0.9283	0.0893	0.1343	0.1629
Book	0.773	0.0326	0.0626	0.8044	0.0461	0.105	0.0872
Film	0.819	0.0258	0.0500	0.8026	0.0411	0.1804	0.0782

Table 4.7: Best overall results: micro average of precision (P), recall (R), F1 measure (F1) and estimated upper bound for recall over all relations. The best normal method is the Stop+Unam+Stat75 training data selection strategy and the MultiLab+Limit75 integration strategy, the best "relaxed" method uses the same strategies for training data selection and information integration and CorefN for co-reference resolution.

Overall, the lowest precision is achieved for *Musical Artist* and the highest for *Book*.

The results for a bigger set of classes in the automatic evaluation confirm the general tendency already observed for the automatic evaluation. The (reimplemented) **Mintz** baseline model has the lowest precision out of all models. This is partly because the amount of available training data for those models is much smaller than for other models. For candidate identification, only entities recognised by Stanford NERC are used and in addition the approach by Mintz et al. (2009) only uses 1% of available negative training data. For other models NER heuristics are used in addition, which do not assign a NE label to instances. As a result, the NE class feature for the relation extractor is missing for all those NEs only detected by NER heuristics, which makes it much more difficult to predict a label. In the Mintz et al. (2009) paper this is solved by using more training data and only training a classifier for relations which have at least 7000 training examples. As Mintz et al. (2009)'s approach and other distant supervision approaches use different corpora and a different evaluation setup, the experiments documented in this chapter cannot be directly compared with those other approaches.

The **Comb** group of models have a much higher precision than the Mintz model. This difference can be explained by the difference in features, but mostly the fact that the Mintz model only uses 1% of available negative training data. The absolute number of correctly recognised property values in the text is about 5 times as high as the Mintz group of features which, again, is due to the fact that Stanford NERC fails to recognise some of the relevant entities in the text.

For the different training data selection methods, **Unam, Stop, Stat and StatRes**, it can be observed that removing some of the ambiguities helps to improve the precision of models, but always at the expense of recall. However, removing too many positive training instances also hurts precision. The highest overall precision is achieved using the Stop+Unam+Stat75 training data selection method.

Although strategically selecting training instances improves precision, the different **information integration** methods tested have a much bigger impact on precision. Allowing multiple labels for predictions (MultiLab) amounts to a significant boost in precision, as well as restricting the maximum number of results per relation. Limit75 leads to a higher precision than LimitMax at a small decrease in recall.

The different models based on the relaxed setting show a surprisingly high precision. They outperform all models in terms of recall, and even increase precision for most classes. The classes they do not increase precision for are "Educational Institution" and "Film", both of which already have a high precision for the normal setting. The NoSub model has the highest recall out of all models based on the relaxed setting, since it is the least restrictive one. However, it also has the lowest precision.

The different co-reference resolution models overall achieve very similar precision and recall. There is a difference in performance between different classes though: the gazetteer-based method outperforms the Stanford Coref model in terms of precision for the classes "Musical Artist", "Business Operation", "Educational Institution" and "Film", whereas the Stanford Coref method outperforms the gazetteer-based method for "Politician", "River" and "Book". Stanford Coref relies on Stanford NER as a pre-processing step. If the latter fails, co-references cannot be resolved.

As analysed in depth in the next two chapters, named entity recognition and classification is easiest for persons and locations (such as politicians and rivers) and more challenging for organisations and especially miscellaneous NEs. Performing such tasks in diverse genres such as the Web genre makes it all the more challenging. NERC failure could be part of the reason Stanford Coref performs better for some entities, whereas gazetteer-based methods perform better for others. This suggests that in the context of Web information extraction for knowledge base population, simple co-reference resolution methods based on synonym gazetteers are equally as effective as supervised co-reference resolution models overall.

The models which perform co-reference resolution have about the same recall as other models, but increase precision by up to 11% depending on the class. The reason those models perform so well is that individual predictions are combined. Even if predicting relations for individual instances is more challenging using co-reference resolution compared to just using sentences which contain mentions of entities explicitly, predictions for some testing instances can be made with a high confidence. This redundancy gained from additional results helps to improve overall precision. However, it is possible that with more testing data, performing co-reference resolution on the testing data might not increase precision. Examining the relationship between using co-reference resolution and increased recall is left for future work.

In general, the availability of test data poses a challenge, which is reflected by the estimated upper bound for recall (see Table 4.4). The upper bound is quite low, depending on the class between 0.035 and 0.23. Using search based methods to retrieve Web pages for training and testing is quite widely used, e.g. Vlachos and Clark (2014b) also use it for gathering a corpus for distant supervision. To increase the upper bound, one strategy could be to just retrieve more pages per query, as Vlachos and Clark (2014b) do. Another option would be to use a more sophisticated method for building search queries, as for instance researched by West et al. (2014). What was not investigated in the context of this thesis is which websites the correct and incorrect extractions are from. It might be that the correct extractions are from more structured sources such as Wikipedia or imdb (see Table 4.2 for a list of training data sources)

As for different relations and classes (see Table 4.7), it can be observed that there is a sizable difference in precision for them. Overall, the lowest precision is achieved for *Musical Artist* and the highest for *Educational Institution*. The reason for this could partly be albums and tracks are only recognised with NER heuristics, but not with Stanford NERC, a problem that is addressed in Chapter 6.

When examining the training set it is further observed that there seems to be a strong correlation between the number of training instances and the precision for that property. This is also an explanation as to why removing possibly ambiguous training instances only improves precision up to a certain point: the classifier is better at dealing with noisy training data than too little training data.

Also, the test data is analysed to try to identify patterns of errors. The two biggest groups of errors are entity boundary recognition and subject identification errors. An example for the first group is the following sentence:

"<s>The Hunt for Red October</s> remains a masterpiece of military <o>fiction</o>."

Although "fiction" would be a correct result in general, the correct property value for this specific sentence would be "military fiction". The NER heuristics suggest both as possible candidates (since both greedy and non-greedy matching are employed, but the relation classifier should only classify the complete noun phrase as a value of *Book:genre*. There are several reasons for this: "military fiction" is more specific than "fiction", and since Freebase often contains the general category ("fiction") in addition to more fine-grained categories, more property values for abstract categories are available to use as instances for training than for more specific categories. Second, the Web corpus also contains more mentions for broader categories than for more specific ones. Third, when annotating training data, positive candidates are not restricted to whole noun phrases, as explained in Section 4.3.2. As a result, if none of the lexicalisations of the entity match the whole noun phrase, but there is a lexicalisation which matches part of the phrase, that is used for training and the classifier learns wrong entity boundaries. Instead of requiring strict matching of entity boundaries, following the ConLL 2003 NERC evaluation guidelines (Tjong Kim Sang and De Meulder, 2003), lenient matches could also be allowed, as e.g. in the context of (Walker et al., 2006). For the ACE evaluation, partial matches are allowed if the head of the NE matches and a certain minimum proportion of characters match. Different criteria for comparing NE anontations are also discussed and compared in Demetriou et al. (2008).

The second big group of errors is that occurrences are classified for the correct relation, but the wrong subject.

"<s>Anna Karenina</s> is also mentioned in <o>R. L. Stine</o>'s Goosebumps series Don't Go To Sleep."

In that example, "R. L. Stine" is predicted to be a property value for *Book:author* for the entity "Anna Karenina". This happens because, at the moment, we do not take into consideration that two entities can be in *more than one* relation. Therefore, the classifier learns wrong, positive weights for certain contexts.

4.6 Discussion

This chapter proposes and evaluates a distantly supervised class-based approach for relation extraction from the Web which strategically selects instances for training, extracts relations across sentence boundaries, and integrates relations for knowledge base population. Previous distantly supervised approaches have been tailored towards extraction from narrow genres, such as news and Wikipedia, and are therefore not fit for Web relation extraction: they fail to identify named entities correctly; they suffer from data sparsity; and they either do not try to resolve noise caused by ambiguity or do so at a significant increase of runtime. They further assume that every sentence may contain any entity in the knowledge base, which is very costly. The goals of this chapter are therefore to:

- improve named entity recognition and classification
- use more testing data by integrating co-reference resolution methods
- experiment with how to combine extractions
- research simple and cheap methods of reducing noise for distant supervision

- propose a setting for distant supervision which uses Web search to identify Web pages containing sentences which are relevant to the relation in question

The research described in this chapter has made a first step towards achieving those goals. Experiments with simple NER heuristics are presented, which are used in addition to a NERC trained for the news genre. Findings for this are that it can especially improve on the number of extractions for non-standard named entity classes such as *MusicalArtist:track* and *MusicalArtist:album*. At the moment, the NER heuristics only recognise, but do not classify NEs. In the following chapters, the goal is to to research distantly supervised named entity classification methods to assist relation extraction.

To overcome data sparsity and increase the number of extractions, co-references are resolved and relations are extracted across sentence boundaries and integrated for knowledge base population. Findings are that extracting relations across sentence boundaries not only increases recall by up to 25% depending on the model, but also increases precision by 8% on average. Moreover, a finding is that while Stanford Coref works well for Freebase classes including "Politician", for other types such as "Film", a gazetteer-based method for co-reference resolution performs better. To populate knowledge bases, different information integration strategies are tested, which differ in performance by 5%.

Further, it is demonstrated that simple, statistical methods to select instances for training can help to improve the performance of distantly supervised Web relation extractors, increasing precision by 3% on average. The performance of those methods is dependent on the type of relation they are applied to and on how many instances there are available for training. Removing too many training instances tends to hurt performance rather than improve it.

One potential downside of using distant supervision for knowledge base population is that it either requires a very large corpus, such as the Web, or a big knowledge base for training. As such, distant supervision itself is an minimally supervised domain-independent approach, but might not necessarily be useful for scenarios for which only a small corpus of documents or only a very small number of relation tuples is available in the knowledge base. For the experiments documented in this chapter, a relatively large part of the knowledge base is used for training, i.e. 1000 entities for training per class, and 10 Web documents per entity and relation. In other experimental setups for distant supervision, only 30 entities, but 300 Web documents per entity and relation are used (Vlachos and Clark, 2014b). It is not just the quantity of documents retrieved that matters, but also the relevance to the information extraction task. Information retrieval for Web relation extraction, i.e. how to formulate queries to retrieve relevant documents for the relation extraction task is something that has already been researched, but not been exploited for distant supervision yet (West et al., 2014).

4.7 Summary

This chapter contains research and experiments addressing six shortcomings of distant supervision.

In order to decrease the noise of training data, methods for **selecting training instances** are researched and documented in this chapter. The methods aim at exploiting background knowledge

from knowledge bases even further to measure ambiguity of terms and then select training data which is very unlikely to be noisy. These methods can be used prior to the learning process, whereas other state of the art methods aim at decreasing the noise of training data as part of the learning process, selecting more suitable contexts or manually improve training data. It would therefore be easy to apply those methods in conjunction with each other as they are complementary. Results indicate that these methods improve precision, but that other factors might play a more important role, such as candidate identification with named entity recognition and classification, acquiring additional test data with co-reference resolution and information integration.

As for **selecting testing instances**, a popular off-the-shelf co-reference resolution approach, Stanford Coref (Lee et al., 2013) is compared against heuristics relying on gender and number gazetteers (Bergsma and Lin, 2006) and gazetteers created from Wikipedia redirection pages and WordNet (Fellbaum, 1998), and also against a baseline which does not try to resolve the subjects of relations. Findings are that co-reference resolution methods for acquiring additional testing data increase performance for knowledge base population. Further, Stanford Coref and the gazetteer-based methods suggested in this chapter show complementary performance. Co-reference resolution methods increase recall by up to 25% depending on the model and increase precision by 8% on average for the setting in this chapter. Those are promising results, however, there are still several open research questions not investigated in this respect. One reason co-reference resolution is helpful is the number of additional extracted relations for knowledge base population. What is left for future research is to study what the exact relation between the number of results per relation candidate and F1 are.

A different **setting** for the evaluation of distant supervision systems is presented in this chapter. An architecture for Web search-based distant supervision is proposed and discussed. The approach uses search queries about instances to retrieve top ranked Web pages. In contrast to existing work, relations can be extracted from a small set of Web pages instead of a large corpus, which should reduce the probability of false predictions for knowledge base population and reduce computational effort. However, no direct empirical evaluation and comparison of this setup is made to other settings.

The **evaluation** setup in this chapter demonstrates that the manual evaluation and an automatic evaluation for knowledge base population show very similar results. It can therefore be concluded that, although an automatic knowledge base population evaluation relies on imprecise annotations, it is suitable to assess the performance of distant supervision methods. This setup is therefore also used in Chapter 6. A **corpus** for Web-based distant supervision has been created which contains sentences with relations of seven popular Freebase types: Musical Artist, Politician, Business Operation, Educational Institution, Film, Book and River. The corpus contains Web pages from over 100 different websites per Freebase type, making it a diverse corpus. The corpus is also used for further experiments documented in Chapter 6.

Different methods for **combining predictions** also substantially help to improve the precision for knowledge base population, namely allowing multiple labels for predictions, as well as restricting the maximum number of results per relation.

Further, the chapter contains early experiments on **named entity recognition** heuristics for

relation extraction to make up for errors caused by an off-the-shelf tool, namely Stanford NER. The heuristics use POS tags and Web-based heuristics, i.e. HTML markup. Named entity recognition and classification is an integral part of relation extraction since relation extraction directly depends on named entity recognition and classification for relation candidate identification, and errors made at this stage are difficult to make up for at the relation extraction stage. NERC is well-researched for the newswire genre, for which corpora and tools are readily available, but less so for more diverse genres such as the Web. As observed in early experiments documented in this chapter, applying supervised NERC approaches trained on newswire to the more diverse Web genre results in NEs not being recognised.

To summarise, this chapter introduces a Web search-based setting for gathering training and testing data for distant supervision and it introduces a new corpus for Web-based distant supervision. Simple methods for discarding noisy training data for distant supervision are proposed and evaluated. These methods are less computationally expensive than previously proposed methods and improve precision for knowledge base population. Relations are extracted from sentences which do not contain the proper name of the subject of the relation, based on existing co-reference resolution approaches and simpler approaches based on synonyms and number and gender gazetteers. Those further improve the precision for knowledge base population. Several simple methods for combining extractions for knowledge base population are evaluated, which take into account how many results to return and if extractions can have multiple labels. Finally, simple methods for increasing the number of relation candidates with NER heuristics are studied since off-the-shelf supervised NER solutions often fail on noisy Web text. The next chapter now studies the problem of named entity recognition and classification in diverse genres in more detail.

Since named entity recognition and classification is such an important part of relation extraction, the next chapter (Chapter 5) documents studies why NERC errors happen, particularly in diverse genres such as the Web, and quantifies these errors.

Chapter 5

Recognition and Classification of Diverse Named Entities

5.1 Introduction

Named entity recognition and classification (*NERC*, short *NER*), the task of recognising and assigning a type to mentions of proper names (named entities, *NEs*) in text, has a long standing tradition, starting from the first MUC challenge in 1987 (Grishman and Sundheim, 1995).

NERC is an integral part of relation extraction since relation extraction directly depends on named entity recognition and classification for relation candidate identification. Unrecognised NEs are a big problem for Web-based distant supervision, as early experiments in the previous chapter have already shown. If suitable corpora are readily available, using trained NERC tools as preprocessing can be successful, but this poses a more substantial problem for the Web genre because standard pre-trained NERC tools do not work well on this type of text.

Generally speaking, NLP methods struggle with noisy text, that is, text that contains spelling errors, abbreviations, dialectical and informal usage and grammatical mistakes (Subramaniam et al., 2009). Such informal communications appear often on Web pages, blogs, tweets or in online chat, or generally speaking any kind of unedited user-generated content. The corpus used for relation extraction experiments in this thesis contains a variety of such content (see Table 4.2): websites with user reviews such as Amazon, Goodreads or Tripadvisor or knowledge exchange websites such as Answers.com.

The prevailing assumption has been that lower NER performance in diverse genres such as social media is due to differences arising from using newswire as training data, as well as from language irregularities (e.g. Ritter et al. (2011)). No prior studies, however, have investigated this quantitatively; for example, it is unknown if this performance drop is really due to a higher proportion of unseen NEs in the social media, or NEs being situated in different kinds of linguistic context.

This chapter quantifies errors made by NERC tools, and analyses and compares NERC approaches on different gold standard corpora to get a better understanding of the NERC task

for different genres[1]. *Domains* and *genres* are two of the characterising distinctions of corpora. Domains typically indicate subject fields such as "politics" or "science", whereas genres help to differentiate between the type of communication such as "broadcast news" or "phone calls", though some definitions and categorisations overlap (Lee, 2001). This chapter focuses on analysing corpora for different genres. Notable genres analysed include Web data and Twitter data, since informal language is even more pronounced in the latter, which has lead to much research on this topic in the past few years (e.g. Baldwin et al. (2015)) and the creation of NERC corpora for Twitter (Finin et al., 2010; Ritter et al., 2011; Rowe et al., 2013).

Existing studies suggest that named entity diversity, a discrepancy between named entities in the training set and the test set, and a diverse context makes NER more difficult (Derczynski et al., 2015b). As mentioned above, user generated content on Web pages or social media contains noise, which means such corpora contain an even more diverse context and more diverse NE mentions. The hypothesis analysed in this chapter is that one of the main reasons for NERC failure is that NERC tools heavily rely on direct lexical matches as cues and have difficulties generalising over a training set to recognise unseen NE mentions in the test corpora. The latter is a challenge arising from *entity drift* (Derczynski et al., 2013; Fromreide et al., 2014) over time. Drift is where unseen NE mentions in user generated content such as Twitter or blogs increase over time since popular topics discussed change. In addition, unseen words increase over time since linguistic conventions change very quickly due to new abbreviations or colloquialisms emerging. In practice, annotated corpora become less and less useful over time (Eisenstein, 2013).

This chapter aims to quantify how NERC diversity impacts different NER methods, by measuring named entity (NE) and context variability, feature sparsity, and their effects on precision, recall and F1.

In particular, the findings indicate that supervised NERC methods struggle to generalise in diverse genres with limited training data. Further, the best predictor for a high NERC performance is found to be the percentage of NEs which appear in both the training and the test corpus.

After studying the NERC task for diverse genres, the chapter discusses what the implications of the findings of this chapter are for relation extraction on Web data and how a successful NERC method for distantly supervised relation extraction for Web pages can be developed (Section 5.6).

Accordingly, the contributions of this chapter lie in investigating the following open research questions:

RQ1 How does NERC performance differ for corpora over different text types/genres?

RQ2 What is the impact of NE diversity on NERC performance?

RQ3 How well do NERC methods perform out-of-genre and what impact do unseen NEs (i.e. those which appear in the test set, but not the training set) have on out-of-genre performance?

RQ4 What is the relationship between Out-of-Vocabulary (OOV) features (unseen features), OOV entities (unseen NEs) and performance?

To ensure representativeness and comprehensiveness, experiments in this chapter are per-

[1]The content of this chapter is based on work which is currently under review with the journal Information Processing & Management (Augenstein et al., 2015a).

formed on key benchmark NER corpora spanning multiple genres, time periods, and corpus annotation methodologies and guidelines. As detailed in Section 5.4.1, the corpora studied are OntoNotes (Hovy et al., 2006), ACE (Walker et al., 2006), MUC 7 (Chinchor, 1998), the Ritter NER corpus (Ritter et al., 2011), the MSM 2013 corpus (Rowe et al., 2013), and the UMBC Twitter corpus (Finin et al., 2010). To eliminate potential bias from the choice of statistical NER approach, experiments are carried out with three differently-biased NER approaches, namely Stanford NER, SENNA and CRFSuite (see Section 5.4.2 for details).

5.2 Related Work

The general problem of machine learning-based named entity recognition has been addressed in the literature over many years, and so there are accompanying analyses. For example, early NER tasks such as the MUC series prompted a statistical analysis (Palmer and Day, 1997). Later, there were major general surveys of existing NER methods, such as (Nadeau and Sekine, 2007). In turn, desiderata and frameworks for general NER have been developed and presented alongside implemented approaches (Ratinov and Roth, 2009).

Throughout the years, it has been important to regularise and avoid overfitting – even in cases where there are large corpora all of the same type and with the same entity classification scheme, such as ACE (Mooney and Bunescu, 2005). This becomes more important as the scope and variety of the text broadens, e.g. when moving from newswire to web text (Whitelaw et al., 2008). Additionally, recall has been a problem, as named entities often seem to have unusual surface forms. They may consist of unusual character sequences for that language (e.g. *Szeged*) or words that individually are typically not NEs, but as phrases are being used as one (e.g. *the White House*). Gazetteers often play a key role in overcoming low NER recall (Mikheev et al., 1999). Research on gazetteer collection has moved from manual assembly (Cunningham et al., 2002), through automatic collection (Kozareva, 2006; Maynard et al., 2009), and now the most recent NER challenges have distributed entity gazetteers (Baldwin et al., 2015) derived from linked data (Bollacker et al., 2008) as part of their baseline systems.

Indeed, the move from ACE and MUC to broader kinds of corpus seem to present existing NER resources and resources with a great deal of difficulty (Maynard et al., 2003). This has led to productive research on domain adaptation, specifically with entity recognition in mind (Daumé, 2007; Wu et al., 2009; Chiticariu et al., 2010). However, in more recent comparisons of NER performance over different corpora with different methods, the older tools tend to simply fail to adapt, even given a fair amount of in-genre data and support (Ritter et al., 2011; Derczynski et al., 2015b). Simultaneously, the value in non-newswire data has rocketed: for example, social media now provides us with a sample of all human discourse, which is unedited and does not follow publishing guidelines, and all in digital format – leading to whole new fields of research opening in computational social science (Hovy et al., 2015; Plank and Hovy, 2015; Preoţiuc-Pietro et al., 2015).

Research on NER for social media content is, accordingly, a highly popular research area (Ritter et al., 2011; Liu et al., 2011; Plank et al., 2014; Derczynski et al., 2015b; Cherry and Guo, 2015),

with multiple recent shared tasks (Rowe et al., 2015; Baldwin et al., 2015). The task is generally cast as a genre adaptation problem from newswire data, integrating the two kinds of data for training (Cherry and Guo, 2015) or including a lexical normalisation step (Han and Baldwin, 2011) to move the problem back to territory more familiar to existing models and methods. Two major perceived challenges are that NEs mentioned in tweets change over time, i.e. entity drift (Derczynski et al., 2013; Fromreide et al., 2014), and that diversity of context makes NER more difficult (Derczynski et al., 2015b).

This chapter takes a new angle on genre adaptation and the progress of named entity recognition research. The idea explored in this chapter is that overfitting has occurred not only at the level of dataset and model, but also through the community's reliance on ageing, low-variety datasets over a long period. Accordingly, analyses of multiple systems are performed, comprising different approaches to statistical NER, over multiple text genres with varying NE and lexical diversity. In line with prior work on analysing NER performance (Palmer and Day, 1997; Derczynski et al., 2015b), Section 5.4 next starts by carrying out corpus analysis and introduces briefly the NER methods used for experimentation. Unlike prior efforts, however, the main objectives of this chapter are to uncover the impact of NE diversity and context diversity on F1, and also to study the relationship between OOV NEs and features and F1 (see Section 5.5 for details).

5.3 Experiments

5.4 Datasets and Methods

5.4.1 Datasets

Since the goal of this study is to compare NER performance on corpora from diverse domains and genres, seven benchmark NER corpora are included, spanning newswire, broadcast conversation, web content, and social media (see Table 5.1 for details). These datasets were chosen such that they have been annotated with the same or very similar entity types, in particular, names of people, locations, and organisations. Thus corpora including only domain-specific entities (e.g. biomedical corpora) were excluded. The choice of corpora was also motivated by their chronological age, i.e. we wanted to ensure a good temporal diversity, in order to study possible effects of entity drift.

Corpora Used

In chronological order, the first corpus included here is MUC 7, which is the last one of the MUC challenges (Chinchor, 1998). This is an important corpus, since the Message Understanding Conference (MUC) was the first one to introduce the NER task in 1995 (Grishman and Sundheim, 1995), with a focus on recognising persons, locations and organisations in newswire text.

A subsequent evaluation campaign was the ConLL 2003 NER shared task (Tjong Kim Sang and De Meulder, 2003), which created gold standard data for newswire in Spanish, Dutch, English and German. The corpus of this evaluation effort is now one of the most popular gold standards for NER, with new NER tools and methods typically reporting performance on that.

Later evaluation campaigns began addressing NER for genres other than newswire, specifically ACE (Walker et al., 2006) and OntoNotes (Hovy et al., 2006). Both of those contain subcorpora in several genres, namely newswire, broadcast news, broadcast conversation, weblogs, and conversational telephone speech. ACE, in addition, contains a subcorpus with usenet newsgroups. The languages covered are English, Arabic and Chinese. A further difference between the ACE and Ontonotes corpora on one hand, and CoNLL and MUC on the other, is that they contain annotations not only for NER, but also for other tasks such as coreference resolution, relation and event extraction and word sense disambiguation. In this chapter, however, we restrict ourselves purely to the English NER annotations, for consistency across datasets.

With the emergence of social media, studying NER performance on this genre gained momentum. So far, there have been no big evaluation efforts, such as ACE and OntoNotes, resulting in substantial amounts of gold standard data. Instead, benchmark corpora were created as part of smaller challenges or individual projects. The first such corpus is the UMBC corpus for Twitter NER (Finin et al., 2010), where researchers used crowdsourcing to obtain annotations for persons, locations and organisations. A further Twitter NER corpus was created by Ritter et al. (2011), which, in contrast to other corpora, contains more fine-grained types defined by the Freebase schema (Bollacker et al., 2008). Next, the Making Sense of Microposts initiative (Rowe et al., 2013) provides single annotated data for named entity recognition on Twitter for persons, locations, organisations and miscellaneous. MSM initiatives from 2014 onwards in addition feature a named entity linking task, but since we only focus on NER here, the 2013 corpus is used.

These corpora are diverse not only in terms of genres and time periods covered, but also in terms of NE types and their definitions. In particular, the ACE and OntoNotes corpora try to model entity metonymy by introducing facilities and geo-political entities (GPEs). Since the rest of the benchmark datasets do not make this distinction, metonymous entities are mapped to a more common entity class (see below).

In order to ensure consistency across corpora, only Person (PER), Location (LOC) and Organisation (ORG) are used in the experiments, and other NE types are mapped to O (no NE). For the Ritter corpus, the 10 entity types are collapsed to 3 as in Ritter et al. (2011). For the ACE and OntoNotes corpora, the following mapping is used: PERSON $\rightarrow$ PER; LOCATION, FACILITY, GPE $\rightarrow$ LOC; ORGANIZATION $\rightarrow$ ORG; all other types $\rightarrow$ O.

Tokens are annotated with BIO sequence tags, indicating that they are the beginning (B) or inside (I) of NE mentions, or outside of NE mentions (O). For the Ritter and ACE 2005 corpora, separate training and test corpora are not publicly available, so we randomly sample 1/3 for testing and use the rest for training. Separate models are then trained on the training parts of each corpus and evaluated on the development (if available) and test parts of the same corpus. If development corpora are available, as they are for ConLL (ConLL Test A) and MUC (MUC 7 Dev), they are not merged with the training corpora for testing, as it was permitted to do in the context of those evaluation challenges.

Corpus	Genre	N	PER	LOC	ORG
MUC 7 Train	Newswire (NW)	552	98	172	282
MUC 7 Dev	Newswire (NW)	572	93	193	286
MUC 7 Test	Newswire (NW)	863	145	244	474
ConLL Train	Newswire (NW)	20061	6600	7140	6321
ConLL TestA	Newswire (NW)	4229	1641	1434	1154
ConLL TestB	Newswire (NW)	4946	1617	1668	1661
ACE NW	Newswire (NW)	3835	894	2238	703
ACE BN	Broadcast News (BN)	2067	830	885	352
ACE BC	Broadcast Conversation (BC)	1746	662	795	289
ACE WL	Weblog (WEB)	1716	756	411	549
ACE CTS	Conversational Telephone Speech (CTS)	2667	2256	347	64
ACE UN	Usenet Newsgroups (UN)	668	277	243	148
OntoNotes NW	Newswire (NW)	52055	12640	16966	22449
OntoNotes BN	Broadcast News (BN)	14213	5259	5919	3035
OntoNotes BC	Broadcast Conversation (BC)	7676	3224	2940	1512
OntoNotes WB	Weblog (WEB)	6080	2591	2319	1170
OntoNotes TC	Telephone Conversations (TC)	1430	745	569	116
OntoNotes MZ	Magazine (MZ)	8150	2895	3569	1686
MSM 2013 Train	Twitter (TWI)	2815	1660	575	580
MSM 2013 Test	Twitter (TWI)	1432	1110	98	224
Ritter	Twitter (TWI)	1221	454	380	387
UMBC	Twitter (TWI)	510	172	168	170

Table 5.1: Corpora genres and number of NEs of different types

Dataset Sizes and Characteristics

Table 5.1 shows what genres the different corpora belong to, the number of NEs and the proportions of NE types per corpus. Sizes of NER corpora have increased over time, from MUC to OntoNotes.

Further, the type distribution varies between corpora: while the ConLL corpus is very balanced and contains about equal numbers of PER, LOC and ORG NEs, other corpora are not. The most imbalanced corpus is the MSM 2013 Test corpus, which contains 98 LOC NEs, but 1110 PER NEs.

This makes it very difficult to compare NER performance across corpora, since performance partly depends on training data size. Since comparing NER performance as such is not the goal of this research, the impact of training data size is illustrated by using learning curves in the next section. NERC performance is shown on trained corpora normalised by size in Table 5.5. For subsequent experiments throughout this chapter, only the original training data sizes are used.

In order to compare corpus diversity across genres, NE and token/type diversity metrics are used (see e.g. Palmer and Day (1997)). Table 5.3 shows the ratios between the number of NEs and the number of unique NEs per corpus, while Table 5.2 reports the token/type ratios. The

Corpus	Genre	Tokens	Types	Ratio	Norm Ratio
MUC 7 Train	NW	8476	2086	4.06	3.62
MUC 7 Dev	NW	9117	1722	5.29	4.79
MUC 7 Test	NW	12960	2895	4.48	3.80
ConLL Train	NW	204567	23624	8.66	**2.91**
ConLL TestA	NW	34392	7815	4.40	**2.62**
ConLL TestB	NW	39474	8428	4.68	**2.64**
ACE NW	NW	66875	8725	7.66	3.40
ACE BN	BN	66534	7630	8.72	3.40
ACE BC	BC	52758	5913	8.92	4.40
ACE WL	WEB	50227	8529	5.89	**3.12**
ACE CTS	CTS	58205	3425	16.99	7.22
ACE UN	UN	82515	8480	9.73	4.49
OntoNotes NW	NW	1049713	42716	24.57	3.69
OntoNotes BN	BN	259347	16803	15.43	3.77
OntoNotes BC	BC	245545	13218	18.58	3.95
OntoNotes WB	WEB	205081	17659	11.61	3.86
OntoNotes TC	TC	110135	5895	18.68	6.98
OntoNotes MZ	MZ	197517	15412	12.82	3.68
MSM 2013 Train	TWI	56722	10139	5.59	3.50
MSM 2013 Test	TWI	32295	6474	4.99	3.66
Ritter	TWI	48864	10587	4.62	**2.78**
UMBC	TWI	7037	3589	1.96	**1.96**

Table 5.2: Token/type ratios and normalised token/type ratios of different corpora

lower those ratios are, the more diverse a corpus is. While token/type ratios also include tokens which are NEs, they are a good measure of broader linguistic diversity.

While these are good metrics, there are other factors which contribute to corpus diversity, including how big a corpus is and how well sampled it is, e.g. if a corpus is only about one story, it should not be surprising to see a high token/type ratio. Therefore, by experimenting on multiple corpora, from different genres and created through different methodologies, the aim is to encompass all those aspects of corpus diversity.

Since the original NE and token/type ratios do not account for corpus size, Tables 5.2 and 5.3 present also the normalised ratios. For those, a number of tokens equivalent to those in the UMBC corpus (7037) (Table 5.2) or, respectively, a number of NEs equivalent to those in the UMBC corpus (506) are selected (Table 5.3). One possibility for sampling would be to sample tokens and NEs randomly. However, this would not reflect the composition of corpora appropriately. Corpora consist of several documents, tweets or blog entries, which are likely to repeat the words or NEs since they are about one story. The difference between bigger and smaller corpora is then that bigger corpora consist of more of those documents, tweets, blog entries, interviews, etc. Therefore,

Corpus	Genre	NEs	Unique NEs	Ratio	Norm Ratio
MUC 7 Train	NW	552	232	2.38	2.24
MUC 7 Dev	NW	572	238	2.40	2.14
MUC 7 Test	NW	863	371	2.33	1.90
ConLL Train	NW	20038	7228	2.77	1.83
ConLL TestA	NW	4223	2154	1.96	**1.28**
ConLL TestB	NW	4937	2338	2.11	**1.31**
ACE NW	NW	3835	1358	2.82	2.13
ACE BN	BN	2067	929	2.22	1.81
ACE BC	BC	1746	658	2.65	1.99
ACE WL	WEB	1716	931	1.84	**1.63**
ACE CTS	CTS	2667	329	8.11	4.82
ACE UN	UN	668	374	1.79	**1.60**
OntoNotes NW	NW	52055	17748	2.93	1.77
OntoNotes BN	BN	14213	3808	3.73	2.58
OntoNotes BC	BC	7676	2314	3.32	2.47
OntoNotes WB	WEB	6080	2376	2.56	1.99
OntoNotes TC	TC	1430	717	1.99	**1.66**
OntoNotes MZ	MZ	8150	2230	3.65	3.16
MSM 2013 Train	TWI	2815	1817	1.55	**1.41**
MSM 2013 Test	TWI	1432	1028	1.39	**1.32**
Ritter	TWI	1221	957	1.28	**1.20**
UMBC	TWI	506	473	1.07	**1.07**

Table 5.3: NE/Unique NE ratios and normalised NE/Unique NE ratios of different corpora

sampling is instead performed by taking the first n tokens for the token/type ratios or the first n NEs for the NEs/Unique NEs ratios.

Looking at the normalised diversity metrics, the lowest NE/Unique NE ratios <= 1.5 (in bold) can be achieved for TWI corpora as well ConLL Test corpora. The former is not surprising since one would expect noise in social media text such as spelling mistakes to also have an impact on how often the same NEs are seen. The latter is more surprising and suggests that the ConLL corpora are well balanced in terms of stories. Also low NE/Unique ratios (<= 1.7) can be observed for ACE WL, ACE UN and OntoNotes TC. Similar to social media text, weblogs, usenet dicussions and telephone conversations also contain a larger amount of noise compared to the traditionally studied newswire genre, hence this is also not a surprising result. Corpora with high NE/Unique NE ratios (> 2.5) are ACE CTS, OntoNotes MZ and OntoNotes BN. These results are also not surprising. The telephone conversations in ACE CTS are all about the same story and MZ and NW data BN data are expected to be more regular due to editing.

The token/type ratios reflect similar trends. Low token/type ratios <= 2.8 (in bold) are observed for the TWI corpora Ritter and UMBC, as well as ConLL Test corpora. Token/type

ratios are also low (<= 3.2) for ConLL Train and ACE WL. Interestingly, ACE UN and MSM Train and Test do not have low token/type ratios although they have low NE/Unique ratios, i.e. in those corpora, many diverse persons, locations and organisations are mentioned, but similar context words are used. Token/type ratios are high (>= 4.4) for MUC7 Dev, ACE BC, ACE CTS, ACE UN and OntoNotes TC. Telephone conversations (TC) having high token/type ratios is unsuprising since they contain many filler words (e.g. "uh", "you know"), as well as NE corpora, which are generally expected to have regular language use.

Furthermore, it is worth pointing out that, especially for the larger corpora, e.g. OntoNotes NW, size normalisation makes a big difference. The normalised NE/Unique NE ratios drop by almost half compared to the unnormalised ratios, and normalised Token/Type ratios drop by up to 85%. This strengthens the argument for size normalisation and also poses the question if low NERC performance for diverse genres is mostly due to the lack of large training corpora. This is investigated further in Section 5.5.1.

Corpus	Genre	NE tokens	O tokens	Density	Norm Density
MUC 7 Train	NW	914	7562	0.11	0.11
MUC 7 Dev	NW	976	8141	0.11	0.10
MUC 7 Test	NW	1624	11336	0.13	0.13
ConLL Train	NW	29450	174171	0.14	0.15
ConLL TestA	NW	6237	28154	0.18	0.18
ConLL TestB	NW	7194	32279	0.18	0.19
ACE NW	NW	7330	59545	0.11	0.11
ACE BN	BN	3555	62979	0.05	**0.06**
ACE BC	BC	3127	49631	0.06	**0.06**
ACE WL	WEB	3227	47000	0.06	0.08
ACE CTS	TC	3069	55136	0.05	**0.06**
ACE UN	UN	1060	81455	0.01	**0.01**
OntoNotes NW	NW	96669	953044	0.09	0.11
OntoNotes BN	BN	23433	235914	0.09	0.08
OntoNotes BC	BC	13148	232397	0.05	0.11
OntoNotes WB	WEB	10636	194445	0.05	**0.06**
OntoNotes TC	TC	1870	108265	0.02	**0.01**
OntoNotes MZ	MZ	15477	182040	0.08	0.11
MSM 2013 Train	TWI	4535	52187	0.08	0.07
MSM 2013 Test	TWI	2480	29815	0.08	0.07
Ritter	TWI	1842	44627	0.04	**0.04**
UMBC	TWI	747	6290	0.11	0.11

Table 5.4: Tag density and normalised tag density, the proportion of tokens with NE tags to all tokens

Lastly, Table 5.4 reports tag density (percentage of tokens tagged as part of a NE), which is another useful metric of corpus diversity that can be interpreted as the information density of a corpus. What can be observed here is that the NW corpora have the highest tag density and generally tend to have higher tag density than corpora of other genres. Corpora with especially

low tag density <= 0.06 (in bold) are the TC corpora, Ritter, OntoNotes WB, ACE UN, ACE BN and ACE BC. As already mentioned, conversational corpora, to which ACE BC also belong, tend to have many filler words, thus it is not surprising that they have a low tag density. There are only minor differences between the tag density and the normalised tag density, since corpus size as such does not impact tag density.

5.4.2 NER Models and Features

The performance of three widely-used supervised statistical approaches to NER are evaluated: Stanford NER[2], SENNA,[3] and CRFSuite.[4]

These systems have contrasting notable attributes. Stanford NER (Finkel et al., 2005) is the most popular of the three, deployed widely in both research and commerce. The system has been developed in terms of both generalising technologies and also specific additions for certain languages. The majority of openly-available additions to Stanford NER, in terms of models, gazetteers, prefix/suffix handling and so on have been for newswire-type text. Named entity recognition and classification is modelled as a sequence labelling task with a first-order a Conditional Random Field (CRF) model (Lafferty et al., 2001).

SENNA (Collobert et al., 2011) is a more recent system for named entity extraction and other NLP tasks. Using word representations and deep learning with deep convolutional neural networks, the general principle for SENNA is to avoid task-specific engineering while also doing well on multiple benchmarks. The approach taken to fit these desiderata is to use representations induced from large unlabeled datasets, including LM2 (introduced in the paper itself) and Brown clusters (Brown et al., 1992; Derczynski and Chester, 2016). The outcome is a flexible system that is readily adaptable, given training data. Although the system is more flexible in general, it relies on learning language models from unlabelled data. For the setup in Collobert et al. (2011) language models are trained for seven weeks on the English Wikipedia, Reuters RCV1 (Lewis et al., 2004) and parts of the WSJ, and results are reported on ConLL 2003. Reuters RCV1 is chosen as unlabelled data because the English ConLL 2003 corpus is created from the Reuters RCV1 corpus. For the experiments described in this chapter, the original language models distributed with SENNA are used and SENNA is evaluated within the DeepNL framework (Attardi, 2015). As such, it is to some degree also biased towards the ConLL 2003 benchmark data.

Finally, the classical NER approach from CRFsuite (Okazaki, 2007) is used. This frames NER as a structured sequence prediction task, using features derived directly from the training text. Unlike the other systems, no external knowledge (e.g. gazetteers and unsupervised representations) are used. This provides a strong basic supervised system, and has not been tuned for any particular domain or genre, thus having potential to reveal more challenging genres without any intrinsic bias.

The feature extractors natively distributed with the NER tools are used: for Stanford NER, the feature set "chris2009" is used without distributional similarity, for SENNA the only available

[2] http://nlp.stanford.edu/projects/project-ner.shtml
[3] https://github.com/attardi/deepnl
[4] https://github.com/chokkan/crfsuite/blob/master/example/ner.py

one is used, for CRFSuite the provided feature extractor without POS or chunking features is used.

These systems are compared against a simple surface form memorisation tagger. The memorisation baseline picks the most frequent NE label for each token sequence as observed in the training corpus. There are two types of ambiguity: one is overlapping sequences, e.g. if both "New York City" and "New York" are memorised as a location. In that case the longest-matching sequence is labeled with the corresponding NE type. The second, type ambiguity occurs when the same textual label refers to different NE types, e.g. "Google" could either refer to the name of a company, in which case it would be labelled as ORG, or to the company's search engine, which would be labelled as O (no NE).

5.5 Experiments

5.5.1 RQ1: NER performance in Different Domains

		CRFSuite			**Stanford NER**			**SENNA**		
	Genre	P	R	F1	P	R	F1	P	R	F1
MUC 7 Dev	NW	62.95	61.3	62.11	68.94	69.06	69	55.82	65.38	60.23
MUC 7 Test	NW	63.4	50.17	56.02	70.91	51.68	59.79	54.35	51.45	52.86
ConLL TestA	NW	66.63	44.62	53.45	70.31	48.1	57.12	72.22	70.75	71.48
ConLL TestB	NW	67.73	43.47	52.95	69.61	44.88	54.58	48.6	48.46	48.53
ACE NW	NW	**49.73**	30.72	37.98	**46.41**	34.19	39.37	46.78	50.45	48.55
ACE BN	BN	56.69	**13.55**	**21.87**	**56.09**	**26.64**	**36.12**	40.07	**36.83**	**38.38**
ACE BC	BC	59.46	29.88	39.77	60.51	40.07	48.21	**39.46**	**41.94**	**40.66**
ACE WL	WEB	65.48	**21.65**	32.54	59.52	**22.57**	**32.73**	53.07	**32.94**	**40.65**
ACE CTS	TC	69.77	**14.61**	**24.15**	74.76	**23.05**	**35.24**	72.36	68	70.11
ACE UN	UN	**20**	**0.41**	**0.81**	**10.81**	**1.65**	**2.87**	**12.59**	**7.44**	**9.35**
OntoNotes NW	NW	53.48	28.42	37.11	64.03	30.45	41.28	**36.84**	51.56	42.97
OntoNotes BN	BN	65.08	55.58	59.96	76.5	57.81	65.86	59.33	66.12	62.54
OntoNotes BC	BC	**49.13**	30.14	37.36	**55.47**	36.56	44.07	**36.33**	50.79	42.36
OntoNotes WB	WEB	**50.41**	22.02	**30.65**	**57.46**	28.83	38.4	51.39	48.16	49.72
OntoNotes TC	TC	67.18	22.82	34.07	65.25	29.44	40.58	59.92	50	54.51
OntoNotes MZ	MZ	58.15	44.1	50.16	74.59	43.84	55.22	54.19	54.64	54.41
MSM 2013 Test	TWI	70.98	36.38	48.11	75.37	38.9	51.31	56.7	60.89	58.72
Ritter	TWI	75.56	25.19	37.78	78.29	29.38	42.73	59.06	46.67	52.14
UMBC	TWI	**47.62**	**10.64**	**17.39**	62.22	**14.81**	**23.93**	**33.15**	**31.75**	**32.43**
Macro Average		58.92	30.82	38.64	63.00	35.36	44.13	49.59	49.17	48.98

Table 5.5: P, R and F1 of NERC with different models evaluated on different testing corpora, trained on corpora normalised by size

The first research question studied in this chapter is whether existing NER approaches generalise well over training data in different genres. In order to answer this, Precision (P), Recall (R)

	Memorisation			**CRFSuite**			**Stanford NER**			**SENNA**		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
MUC 7 Dev	44.35	27.4	33.87	63.49	61.82	62.64	69.98	70.1	70.04	55.12	73.43	62.97
MUC 7 Test	50	18.75	27.27	63.43	50.64	56.31	71.11	51.97	60.05	53.84	60.95	57.17
ConLL TestA	61.15	35.67	45.06	91.51	88.22	89.84	93.09	91.39	92.23	91.73	93.05	92.39
ConLL TestB	54.36	23.01	32.33	87.24	82.55	84.83	88.84	85.42	87.1	85.89	87	86.44
ACE NW	48.25	31.59	38.18	57.11	44.43	49.98	55.43	48.06	51.48	53.45	50.54	51.95
ACE BN	34.24	20.85	25.92	55.26	22.37	31.85	53.61	33.94	41.57	51.15	50.84	50.99
ACE BC	48.98	32.05	38.75	59.04	46.01	51.72	59.41	50.93	54.84	52.22	47.88	49.96
ACE WL	45.26	5.63	10.01	62.74	21.65	32.2	59.72	22.18	32.34	51.03	22.83	31.55
ACE CTS	79.73	17.2	28.3	80.05	32.04	45.76	81.89	39.59	53.38	75.68	67.22	71.2
ACE UN	12.29	11.93	12.11	20	0.41	0.81	13.51	2.07	3.58	22.22	1.65	3.08
OntoNotes NW	39.01	31.49	34.85	82.19	77.35	79.7	84.89	80.78	82.78	79.37	76.76	78.04
OntoNotes BN	18.32	32.98	23.55	86.51	80.59	83.44	88.33	83.42	85.8	84.69	83.7	84.2
OntoNotes BC	17.37	24.08	20.18	75.59	65.26	70.04	76.34	69.02	72.5	70.38	73.4	71.86
OntoNotes WB	52.61	29.27	37.62	64.73	45.52	53.45	68.62	54.13	60.52	63.94	61.3	62.59
OntoNotes TC	6.48	16.55	9.32	65.26	32.4	43.31	68.82	44.6	54.12	73.45	57.84	64.72
OntoNotes MZ	44.56	31.12	36.64	79.87	74.27	76.97	82.07	79.32	80.67	74.42	76.23	75.31
MSM 2013 Test	20.51	7.84	11.35	83.08	56.91	67.55	83.64	60.68	70.34	70.89	70.74	70.81
Ritter	50.81	15.11	23.29	76.36	31.11	44.21	80.57	34.81	48.62	67.43	43.46	52.85
UMBC	76.92	5.29	9.9	47.62	10.64	17.39	62.22	14.81	23.93	33.15	31.75	32.43
Macro Average	42.38	21.99	26.24	68.48	48.64	54.84	70.64	53.54	59.26	63.69	59.50	60.55

Table 5.6: P, R and F1 of NERC with different models trained on original corpora

and F1 metrics are reported on size normalised corpora (Table 5.5) and original corpora (Tables 5.6 and 5.7). The reason for size normalisation is to make results comparable across corpora. For size normalisation, the training corpora are downsampled to include the same number of NEs as the smallest corpus, UMBC. For that, sentences are selected from the beginning of the train part of the corpora so that they include the same number of NEs as UMBC. Other ways of downsampling the corpora would be to select the first n sentences or the first n tokens, where n is the number of sentences in the smallest corpus. The reason that the number of NEs, which represent the number of positive training examples, are chosen for downsampling the corpora is that the number of positive training examples have a much bigger impact on learning than the number of negative training examples. For instance, Forman and Cohen (2004) study topic classification performance for small corpora and sample from the Reuters corpus, among others. They find that adding more negative training data gives little to no improvement, whereas adding positive samples drastically improves performance.

In Table 5.5 with size normalised precision (P), recall (R), and F1-Score (F1), the lowest 5 P, R and F1 values per method (CRFSuite, Stanford NER, SENNA) are in bold and results for all corpora are summed up with macro average.

	Memorisation			CRFSuite			Stanford NER			SENNA		
Corpus	PER	LOC	ORG	PER	LOC	ORG	PER	LOC	ORG	PER	LOC	ORG
MUC 7 Dev	3.67	28.3	43.87	58.54	63.83	63.03	57.61	75.44	70.44	60.83	62.99	63.8
MUC 7 Test	16.77	51.15	9.56	52.87	59.79	55.33	61.09	72.8	50.42	79.08	61.38	47.4
ConLL TestA	34.88	49.11	55.66	92	91.92	84	94.31	93.85	87.09	95.14	94.65	85.55
ConLL TestB	12.8	40.52	41.71	87.45	87.09	79.78	90.47	89.21	81.45	91.42	89.35	78.78
ACE NW	0	52.12	0	37.47	57.35	33.89	39.92	58.46	38.84	49.72	57.89	30.99
ACE BN	13.37	41.26	18	35.58	35.36	14.81	45.86	41.56	31.09	55.74	55.02	26.73
ACE BC	0	62.64	0	44.32	64.07	23.19	49.34	65.24	30.77	44.98	61.81	18.92
ACE WL	1.44	33.06	0	39.01	37.5	10.97	37.41	42.91	7.96	37.75	40.15	5.96
ACE CTS	18.62	63.77	0	46.28	46.64	0	52.67	60.56	0	75.31	47.35	26.09
ACE UN	0	14.92	7.55	3.08	0	0	0	6.17	0	5.48	2.78	0
OntoNotes NW	20.85	55.67	14.85	84.75	82.39	74.82	86.39	85.62	78.47	80.82	85.71	69.08
OntoNotes BN	15.37	27.52	12.59	88.67	86.4	66.64	90.98	88.88	68.78	90.06	87.46	66.24
OntoNotes BC	10.92	24.89	10.9	69.13	79.28	52.26	75.54	79.04	54.48	74.13	81.02	50.98
OntoNotes WB	26.2	54.86	5.71	50.17	67.76	22.33	60.28	72.05	30.21	66.63	73.15	31.03
OntoNotes TC	19.67	7.71	0	40.68	50.38	8.16	54.24	58.43	18.87	67.06	68.39	7.84
OntoNotes MZ	21.39	58.76	4.55	83.49	80.44	52.86	86.44	84.65	56.88	83.92	78.06	48.25
MSM 2013 Test	4.62	44.71	26.14	75.9	43.75	24.14	78.46	42.77	31.37	81.04	45.42	28.12
Ritter	15.54	35.11	20.13	43.19	54.08	33.54	48.65	57	37.97	63.64	61.47	22.22
UMBC	5.97	22.22	0	8.33	32.61	6.06	25.32	39.08	2.94	33.12	52.1	6.38
Macro Average	12.74	40.44	14.27	54.78	58.98	37.15	59.74	63.88	40.95	65.05	63.48	37.60

Table 5.7: F1 per NE type with different models trained on original corpora

Comparing the different methods, it can be observed that the highest F1 results are achieved with SENNA, followed by Stanford NER and CRFSuite. SENNA has a balanced P and R, which can be explained by the use of word embeddings as features, which help with the unseen word problem. For Stanford NER as well as CRFSuite, which do not make use of embeddings, recall is about half of precision. These findings are in line with other work reporting the usefulness of word embeddings and deep learning for a variety of NLP tasks and domains (Socher et al., 2011; Glorot et al., 2011; Bengio, 2012). With respect to individual corpora, the ones where SENNA outperforms other methods by a large margin (>= 13 points in F1) are ConLL Test A, ACE CTS and OntoNotes TC. The first is not surprising since this is the genre SENNA was tuned to in the original publication. The second is more unexpected and could be due to those corpora containing a disproportional amount of PER and LOC NEs compared to ORG NEs, which are easier to tag correctly, as can be seen in Table 5.7, where F1 of NERC methods is reported on the original training data.

The hypothesis that CRFSuite is less tuned for NW corpora and might therefore have a more balanced performance across genres does not hold. Results with CRFSuite for every corpus are worse than the results for that corpus with Stanford NER.

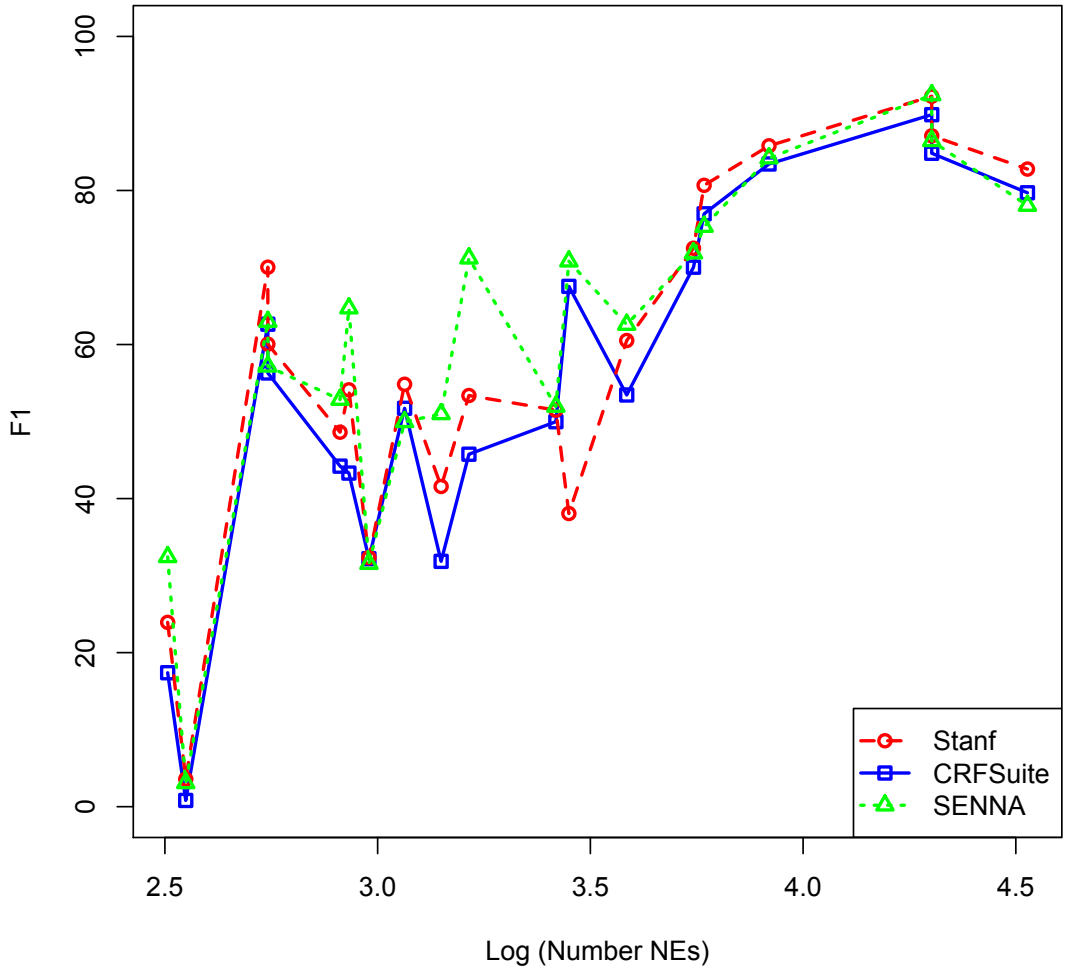

Figure 5.1: F1 of different NER methods with respect to corpus size, measured in log of number of NEs

As for performance on different corpora, even with size normalisation it can be observed that F1 scores vary widely across corpora, e.g. for SENNA ranging from 9.35% F1 (ACE UN) to 71.48% (ConLL Test A). The lowest results are consistently observed for the ACE subcorpora, UMBC, and OntoNotes BC and WB. The highest results are obtained on the ConLL Test A corpus, OntoNotes BN and MUC 7 Dev. This to some degree confirms our hypothesis that NER systems achieve higher performance on NW than on corpora from other genres, probably due to many researchers using them as benchmarks for tuning their approaches. Looking at the TWI corpora which previous work has reported as being challenging, the results are not as clear as anticipated originally. Although results for UMBC are among the lowest, results for MSM 2013 and Ritter are

in the same range or even higher than those on NW datasets. This begs the question whether low results for TWI corpora reported in previous studies were mostly due to the lack of sufficiently large in-genre (i.e. Twitter) training data.

When results on the normalised datasets are compared against those obtained using the original training corpora, TWI results are lower than those for most OntoNotes corpora and ConLL test corpora, mostly due to lower recall. Other corpora with noticably lower NERC performance are ACE UN and WEB corpora. This confirms the hypothesis that social media and Web corpora are amongst the more challenging for NERC, but as mentioned above, part of the reason for this is their small size.

The CoNLL results, on the other hand, are the highest across all corpora irrespective of the NERC method. What is very interesting to see is that they are much higher than the results on the biggest training corpus, OntoNotes NW. For instance, SENNA has an F1 of 78.04 on OntoNotes, compared to an F1 of 92.39 and 86.44 for ConLL Test A and Test B respectively. So even though OntoNotes NW is more than twice the size of ConLL in terms of NEs (see Table 5.3), NERC performance is much higher on ConLL. NERC performance with respect to training corpus size is represented in Figure 5.1. The latter figure confirms that although there is some dependency between corpus size and F1, the variance between results on comparably sized corpora is big. This strengthens the argument that there is a need for experimental studies, such as those reported below, to find out what, apart from corpus size, impacts NERC performance.

Another set of results presented in Table 5.6 are those of the simple NERC memorisation baseline. It can be observed that corpora with a low F1 for NERC methods, such as UMBC and ACE UN, also have a low memorisation performance. For corpora with high NERC performance this holds again, but not for all corpora, e.g. for ConLL Test A and OntoNotes NW, both memorisation performance and NERC performance is high, but memorisation performance for OntoNotes BC is only average.

When NERC results are compared to the corpus diversity statistics (i.e. NE/Unique NE ratios, token/type ratios, and tag density), the strongest predictor for F1 is tag density. There is a loose correlation between high F1 and high tag density, whereas for NE/unique ratios and token/type ratios, no such correlations can be observed. However, tag density is also not an absolute predictor for NERC performance. While NW corpora, which have a high NERC performance also have a high tag density, corpora of other genres with high tag density do not necessarily have a high F1.

In summary, observations in this section are that NERC approaches perform particularly well on the ConLL corpus, and, in general, better on NW corpora than on most other genres. However, normalising corpora by size results in more noisy data such as TWI and WEB data achieving similar results to NW corpora. Therefore, one conclusion is that increasing the amount of available in-genre training data will likely result in improved NERC performance. Corpus size, however, is not an absolute predictor, since NW corpora larger than the ConLL dataset still achieve lower NERC performance. A high tag density is a good, but also not absolute, predictor for high F1. What we found to be a good predictor for a high F1 is a high memorisation performance. Inspired by those findings, the next section will take a closer look at the impact of seen and unsen NEs on NER performance.

5.5.2 RQ2: Impact of NE Diversity

Corpus	Genre	Unseen	Seen	Proportion Unseen
MUC 7 Dev	NW	348	224	0.608
MUC 7 Test	NW	621	241	0.720
ConLL TestA	NW	1485	2744	0.351
ConLL TestB	NW	2453	2496	0.496
ACE NW	NW	549	662	0.453
ACE BN	BN	365	292	0.556
ACE BC	BC	246	343	0.418
ACE WL	WEB	650	112	0.853
ACE CTS	TC	618	410	0.601
ACE UN	UN	274	40	0.873
OntoNotes NW	NW	8350	10029	0.454
OntoNotes BN	BN	2427	3470	0.412
OntoNotes BC	BC	1147	1003	0.533
OntoNotes WB	WEB	1390	840	0.623
OntoNotes TC	TC	486	88	0.847
OntoNotes MZ	MZ	1185	1112	0.516
MSM 2013 Test	TWI	992	440	0.693
Ritter	TWI	302	103	0.746
UMBC	TWI	176	13	0.931

Table 5.8: Proportion of unseen entities in different test corpora

Unseen NEs are those with surface forms present only in the test, but not training data, whereas *seen* NEs are those also encountered in the training data. As discussed previously, the ratio between those two measures is an indicator of corpus NE diversity.

Table 5.8 shows how the number of unseen NEs per test corpus relates to the total number of NEs per corpus. The proportion of unseen forms varies widely by corpus, ranging from 0.351 (ACE NW) to 0.931 (UMBC). As expected there is a correlation between corpus size and percentage of unseen NEs, i.e. smaller corpora such as MUC and UMBC tend to contain a larger proportion of unseen NEs than bigger corpora such as ACE NW. Similarly to the token/type ratios listed in Table 5.2, it can be observed that TWI and WEB corpora have a higher proportion of unseen entities.

As can be seen from Table 5.6, corpora with a low percentage of unseen NEs (e.g. ConLL Test A and OntoNotes NW) tend to have high NERC performance, whereas corpora with high percentage of unseen NEs (e.g. UMBC) tend to have low NERC performance. This seems to suggest that NERC approaches struggle with recognising and classifying unseen NEs correctly. Therefore, next we examine NERC performance for unseen and seen NEs separately.

What becomes clear from the macro averages in Table 5.9[5] is that F1 on unseen NEs is

[5]Note that the performance over unseen and seen entities in Table 5.9 does not add up to the performance reported in Table 5.6 because performance in Table 5.9 is only reported on positive test samples.

		CRFSuite			Stanf			SENNA			All		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
MUC 7 Dev	Seen	96.00	85.71	90.57	98.65	98.21	98.43	90.48	93.3	91.87	95.04	92.41	93.62
	Unseen	65.22	47.41	54.91	75.31	52.59	61.93	59.54	60.06	59.8	66.69	53.35	58.88
MUC 7 Test	Seen	87.02	74.79	80.44	97.81	92.53	95.10	82.81	87.6	85.14	89.21	84.97	86.89
	Unseen	59.95	41.22	48.85	61.93	37.2	46.48	51.57	50.24	50.9	57.82	42.89	48.74
ConLL TestA	Seen	96.79	94.46	95.61	97.87	97.34	97.61	97.34	96.55	96.94	97.33	96.12	96.72
	Unseen	86.34	79.19	82.61	87.85	81.82	84.73	95.32	92.43	93.85	89.84	84.48	87.06
ConLL TestB	Seen	93.70	89.98	91.80	96.07	94.07	95.06	94.3	91.77	93.02	94.69	91.94	93.29
	Unseen	85.71	76.76	80.99	86.76	79.05	82.72	91.69	87.32	89.46	88.05	81.04	84.39
ACE NW	Seen	97.28	64.80	77.79	96.05	69.79	80.84	93.3	63.14	75.32	95.54	65.91	77.98
	Unseen	57.48	22.40	32.24	56.05	25.32	34.88	63.49	36.43	46.3	59.01	28.05	37.81
ACE BN	Seen	93.65	40.41	56.46	94.29	67.81	78.88	91.32	68.49	78.28	93.09	58.90	71.21
	Unseen	66.04	9.59	16.75	47.44	10.14	16.7	71.57	40	51.32	61.68	19.91	28.26
ACE BC	Seen	90.76	65.89	76.35	91.01	70.85	79.67	88.11	62.68	73.25	89.96	66.47	76.42
	Unseen	62.82	19.92	30.25	62.89	24.8	35.57	57.5	28.05	37.7	61.07	24.26	34.51
ACE WL	Seen	89.47	60.71	72.34	96.67	77.68	86.14	91.49	38.39	54.09	92.54	58.93	70.86
	Unseen	75.76	15.38	25.58	61.03	12.77	21.12	62.21	20.77	31.14	66.33	16.31	25.95
ACE CTS	Seen	97.38	45.37	61.90	98.48	63.17	76.97	96.35	64.39	77.19	97.40	57.64	72.02
	Unseen	95.42	23.66	37.92	92.55	24.11	38.25	96.43	69.9	81.05	94.80	39.22	52.41
ACE UN	Seen	0.00	0.00	0.00	0	0	0	100	1.53	3.02	33.33	0.51	1.01
	Unseen	100.00	0.51	1.02	62.5	1.82	3.55	0	0	0	54.17	0.78	1.52
OntoNotes NW	Seen	95.18	90.44	92.75	96.88	93.98	95.4	73.12	65.76	69.24	88.39	83.39	85.80
	Unseen	73.43	63.00	67.81	76.17	65.8	70.6	96.88	93.98	95.4	82.16	74.26	77.94
OntoNotes BN	Seen	95.60	90.86	93.17	96.75	94.5	95.61	81.76	73.34	77.32	91.37	86.23	88.70
	Unseen	82.67	67.24	74.16	83.45	68.97	75.52	96.75	94.5	95.61	87.62	76.90	81.76
OntoNotes BC	Seen	95.29	88.83	91.95	93.85	88.24	90.96	64.27	59.11	61.58	84.47	78.73	81.50
	Unseen	70.91	47.60	56.96	74.82	55.19	63.52	93.85	88.24	90.96	79.86	63.68	70.48
OntoNotes WB	Seen	91.96	81.57	86.45	94.01	89.64	91.77	63.75	47.73	54.59	83.24	72.98	77.60
	Unseen	58.97	26.49	36.56	64.86	34.39	44.95	94.01	89.64	91.77	72.61	50.17	57.76
OntoNotes TC	Seen	94.03	56.25	70.39	94.81	82.95	88.48	80.2	51.73	62.89	89.68	63.64	73.92
	Unseen	70.79	27.27	39.38	74.8	37.86	50.27	94.81	82.95	88.48	80.13	49.36	59.38
OntoNotes MZ	Seen	95.24	88.89	91.95	99.09	97.75	98.42	71.31	62.86	66.82	88.55	83.17	85.73
	Unseen	75.44	57.95	65.55	80.23	64.05	71.23	99.09	97.75	98.42	84.92	73.25	78.40
MSM 2013 Test	Seen	92.40	69.09	79.06	91.73	78.18	84.42	84.22	69.96	76.43	89.45	72.41	79.97
	Unseen	87.21	52.22	65.32	87.08	54.33	66.91	91.73	78.18	84.42	88.67	61.58	72.22
Ritter	Seen	100.00	65.05	78.82	98.8	79.61	88.17	100	68.93	81.61	99.60	71.20	82.87
	Unseen	79.73	19.54	31.38	76.62	19.54	31.13	78.17	36.75	50	78.17	25.28	37.50
UMBC	Seen	100.00	23.08	37.50	100	53.85	70	90	69.23	78.26	96.67	48.72	61.92
	Unseen	59.38	10.86	18.36	66.67	12.5	21.05	52.78	32.39	40.14	59.61	18.58	26.52
Macro Average	Seen	89.57	67.17	75.02	91.20	78.43	83.79	86.01	65.08	71.41	88.92	70.22	76.74
	Unseen	74.38	37.27	45.61	72.58	40.12	48.48	76.18	62.08	67.20	74.38	46.49	53.76

Table 5.9: P, R and F1 of NERC with different models of unseen and seen NEs

significantly lower than F1 on seen NEs for all three NERC approaches. This is mostly due to recall on unseen NEs being lower than that on seen NEs. In particular, Stanford NER and CRFSuite have almost 50% lower recall on unseen NEs compared to seen NEs. Out of the three different approaches, SENNA is the one with the smallest difference between F1 on seen and unseen NEs, and in fact for many corpora has a lower F1 for seen NEs than Stanford NER. This is because SENNA's features are based on word embeddings, which helps a bit with generalisation to unseen NEs, while at the same time decreasing F1 for seen NEs. Although SENNA appears to be better at generalising than Stanford NER and CRFSuite with simple features, there is still a sizable difference between F1 of seen NEs and unseen NEs. The difference in macro average F1 between unseen and seen NEs for SENNA is 21.77, whereas it is 29.41 for CRFSuite and 35.68 for Stanford NER.

The fact that F1 on unseen entities is significantly lower than F1 on seen NEs partly explains what we observed in the previous section, i.e., that corpora with a high proportion of unseen entities, such as the ACE WL corpus, have an overall lower F1 than corpora of a similar size from other genres, such as the ACE BC corpus (F1 of 30 compared to 50, see Table 5.6).

However, even though the F1 of seen NEs is higher than the F1 of unseen NEs, there is still a significant difference of F1 between seen NEs in corpora of different sizes and genres. For instance, F1 of seen NEs in ACE WL, averaged over the three different approaches, is 70.86, whereas the F1 of seen NEs in the less diverse ACE BC corpus is 76.42. Interestingly, F1 of seen NEs in the TWI corpora MSM and Ritter, averaged over the three different methods, is around 80, whereas the F1 of ACE corpora, which are of similar size, is around 70. Amongst the three smallest corpora (UMBC, MUC Dev and MUC Test), UMBC F1 on seen NEs averaged over the NER approaches is significantly lower (61.92 vs. 93.62 and 86.89 respectively).

To summarise, NE diversity explains a large part of the F1 differences of NER approaches between genres, but not all of it. F1 of only seen NEs is significantly and consistently higher than that of unseen NEs in different corpora, which is mostly due to a lower recall. However, there are still significant F1 differences of seen NEs in different corpora. This means that NE diversity does not account for all of the difference of F1 between corpora of different genres.

5.5.3 RQ3: Out-Of-Genre NER Performance and Memorisation

As the experiments reported above demonstrated and also in line with related work, NERC performance varies across genres, while also being influenced by the size of the available in-genre training data.

Prior work on transfer learning and domain adaptation, e.g. Daumé (2007) has aimed at increasing performance in genres and domains where only small amounts of training data are available. This is achieved by adding out-of domain data from domains where larger amounts of training data exist. For domain adaptation to be successful, however, the source domain needs to be similar to the target domain, i.e. if there is no or very little overlap in terms of contexts of the training and testing instances, the model does not learn any additional helpful weights.

In particular, prior work (Sutton and McCallum, 2005) has reported improving F1 by around 6% through adaptation from the ConLL to the ACE dataset. However, transfer learning becomes

	Memorisation			CRFSuite			Stanf			SENNA		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
MUC 7 Dev	38.24	20.89	27.02	54.27	50.09	52.09	57.01	55.42	56.21	50	59.97	54.53
MUC 7 Test	47.45	24.43	32.25	65.54	49.36	56.31	69.46	54.81	61.27	56.37	55.85	56.11
ConLL TestA	53.14	22.36	31.48	67.12	38.57	48.99	69.22	48.27	56.88	68.62	58.68	63.26
ConLL TestB	55.85	22.49	32.07	67.94	36.41	47.41	67.99	44.11	53.51	64.61	51.94	57.58
ACE NW	29.52	28.48	28.99	40.45	47.4	43.65	40.67	49.46	44.63	41.47	54	46.92
ACE BN	1.49	0.15	0.28	0	0	0	0	0	0	36.7	6.09	10.44
ACE BC	24.42	25.13	24.77	36.06	45.67	40.3	32.73	45.84	38.19	33.37	50.93	40.32
ACE WL	25.45	16.54	20.05	40.53	38.45	39.46	41.39	41.34	41.37	41.48	45.01	43.17
ACE CTS	68.31	25.58	37.23	26.28	16.94	20.6	35.93	22.47	27.65	24.69	23.05	23.84
ACE UN	8.07	27.69	12.5	9.76	40.08	15.7	10.48	42.56	16.82	9.95	49.59	16.57
OntoNotes BN	36.97	26.06	30.57	47.77	68.57	56.31	49.49	46.48	47.94	48.43	46.7	47.55
OntoNotes BC	33.68	24.21	28.17	72.24	64.74	68.29	72.69	66.47	69.44	69.49	70.88	70.18
OntoNotes WB	47.45	31.23	37.67	59.14	53.81	56.35	63.88	60.58	62.19	57.04	57.94	57.49
OntoNotes TC	54.15	28.4	37.26	60.88	48.26	53.84	65.09	60.1	62.5	57.79	62.02	59.83
OntoNotes MZ	40.38	20.1	26.84	47.75	64.05	54.71	51.31	41.05	45.61	43.23	39.05	41.03
MSM 2013 Test	14.87	5.8	8.34	41.29	23.32	29.81	49.2	32.19	38.92	16.81	37.85	23.28
Ritter	42.34	11.6	18.22	35.34	24.69	29.07	37.07	26.91	26.91	27.09	36.79	31.2
UMBC	52.27	12.17	19.74	44.71	20.21	27.84	59.09	27.51	37.55	31.39	22.75	26.38
Macro Average	35.48	19.65	23.87	43.00	38.45	38.99	45.93	40.29	41.45	40.98	43.64	40.51

Table 5.10: Out of genre performance: F1 of NERC with different models

more difficult if the target genre is very noisy or, as mentioned already, too different from the source genre. Locke and Martin (2009) unsuccessfully tried to adapt the ConLL 2003 corpus to a Twitter corpus spanning several topics. They found that hand-annotating a Twitter corpus consisting of 24,000 tokens performs better on new Twitter data than their transfer learning efforts with the ConLL 2003 corpus.

This section explores baseline out-of-domain NERC performance without genre adaptation; what percentage of NEs are seen if there is a difference between the the training and the testing genres; and how the difference in performance on unseen and seen NEs compares to in-genre performance.

The source genre for the experiments here is newswire, where we use the classifier trained on the biggest NW corpus investigated in this study, i.e. Ontonotes NW. That classifier is then applied to all other corpora. The rationale is to test how suitable such a big corpus would be for improving Twitter NER, for which only small training corpora are available.

Results for out-of-genre performance are reported in Table 5.10. The highest F1 performance, and a very similar performance to the in-genre setting (Table 5.7), is achieved on the OntoNotes BC corpus. This is unsurprising as it belongs to a similar genre as the training corpus (broadcast conversation) and the data was collected in the same time period and annotated using the same

		CRFSuite			**Stanf**			**SENNA**			**All**		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1
MUC 7 Dev	Seen	81.25	55.15	65.70	82.1	56.97	67.26	86.21	68.18	76.14	83.19	60.10	69.70
	Unseen	63.40	50.83	56.42	72.22	59.09	65.00	64.79	57.02	60.66	66.80	55.65	60.69
MUC 7 Test	Seen	81.25	54.93	65.55	79.15	52.72	63.29	82.43	61.37	70.36	80.94	56.34	66.40
	Unseen	65.37	50.55	57.01	77.78	65.03	70.83	62.71	60.66	61.67	68.62	58.75	63.17
ConLL TestA	Seen	72.49	35.79	47.92	72.33	44.78	55.31	78.77	58.77	67.31	74.53	46.45	56.85
	Unseen	79.32	49.63	61.06	82.61	60.9	70.12	76.96	65.2	70.59	79.63	58.58	67.26
ConLL TestB	Seen	74.72	35.97	48.56	73.3	43.08	54.27	74.32	52.77	61.71	74.11	43.94	54.85
	Unseen	75.38	42.39	54.27	76.18	53.04	62.54	68.76	56.03	61.75	73.44	50.49	59.52
ACE NW	Seen	76.21	50.45	60.71	79.32	54.03	64.28	86.07	61.72	71.89	80.53	55.40	65.63
	Unseen	46.70	46.05	46.37	45.18	45.81	45.50	43.38	47.21	45.21	45.09	46.36	45.69
ACE BN	Seen	0.00	0.00	0.00	0	0	0.00	96.67	8.19	15.1	32.22	2.73	5.03
	Unseen	0.00	0.00	0.00	0	0	0.00	36.11	4.29	7.67	12.04	1.43	2.56
ACE BC	Seen	82.11	52.65	64.16	82.43	53.82	65.12	88.98	61.76	72.92	84.51	56.08	67.40
	Unseen	39.92	38.15	39.01	42.44	40.56	41.48	41.25	42.57	41.9	41.20	40.43	40.80
ACE WL	Seen	66.00	41.04	50.61	68.82	45.02	54.44	48.2	44.72	64.30	61.01	43.59	56.45
	Unseen	45.79	37.78	41.40	47.39	40.28	43.54	79.49	53.98	46.40	57.56	44.01	43.78
ACE CTS	Seen	91.75	46.55	61.76	82.74	41.59	55.35	87.13	61.55	72.14	87.21	49.90	63.08
	Unseen	54.69	49.30	51.85	58.46	53.52	55.88	54.41	52.11	53.24	55.85	51.64	53.66
ACE UN	Seen	74.51	44.71	55.88	75.93	48.24	58.99	90.99	59.41	71.89	80.48	50.79	62.25
	Unseen	37.50	29.17	32.81	43.48	27.78	33.90	33.93	26.39	29.69	38.30	27.78	32.13
OntoNotes BN	Seen	63.92	53.09	58.00	66.06	56.71	61.03	66.8	58.73	62.50	65.59	56.18	60.51
	Unseen	35.42	32.42	33.85	36.39	34.33	35.33	34.13	32.31	33.20	35.31	33.02	34.13
OntoNotes BC	Seen	84.83	66.05	74.27	86.41	70.08	77.39	87.58	75.85	81.29	86.27	70.66	77.65
	Unseen	76.74	65.54	70.70	82	72.39	76.90	71.95	68.02	69.93	76.90	68.65	72.51
OntoNotes WB	Seen	75.44	58.07	65.62	79.64	65.23	71.71	79.75	64.08	71.06	78.28	62.46	69.46
	Unseen	61.37	47.93	53.82	67.89	54.47	60.44	55.22	49.4	52.15	61.49	50.60	55.47
OntoNotes TC	Seen	71.33	48.89	58.02	76.19	62.9	68.91	84.57	70.02	76.61	77.36	60.60	67.85
	Unseen	67.72	51.50	58.50	75.57	59.28	66.44	58.7	48.5	53.11	67.33	53.09	59.35
OntoNotes MZ	Seen	64.84	46.61	54.24	64.34	48.23	55.13	61.7	46.53	53.05	63.63	47.12	54.14
	Unseen	41.40	28.93	34.06	49.7	32.63	39.40	38.92	30.43	34.16	43.34	30.66	35.87
MSM 2013 Test	Seen	58.90	19.24	29.01	56.25	22.15	31.78	57.08	30.65	39.88	57.41	24.01	33.56
	Unseen	70.30	35.33	47.03	73.5	45.89	56.50	59.12	48.02	53.00	67.64	43.08	52.18
Ritter	Seen	62.75	25.20	35.96	58.77	26.38	36.41	79.69	40.16	53.40	67.07	30.58	41.92
	Unseen	58.90	28.48	38.39	56.47	31.79	40.68	62.5	39.74	48.58	59.29	33.34	42.55
UMBC	Seen	60.53	20.18	30.26	75	26.09	38.71	72.34	29.57	41.98	69.29	25.28	36.98
	Unseen	60.61	27.03	37.38	73.53	33.78	46.30	38.3	24.32	29.75	57.48	28.38	37.81
Macro Average	Seen	69.05	41.92	51.46	69.93	45.45	54.41	78.29	53.00	62.42	72.42	46.79	56.10
	Unseen	54.47	39.50	45.22	58.93	45.03	50.60	54.48	44.79	47.37	55.96	43.11	47.73

Table 5.11: Out of genre performance for unseen vs seen NEs: F1 of NERC with different models

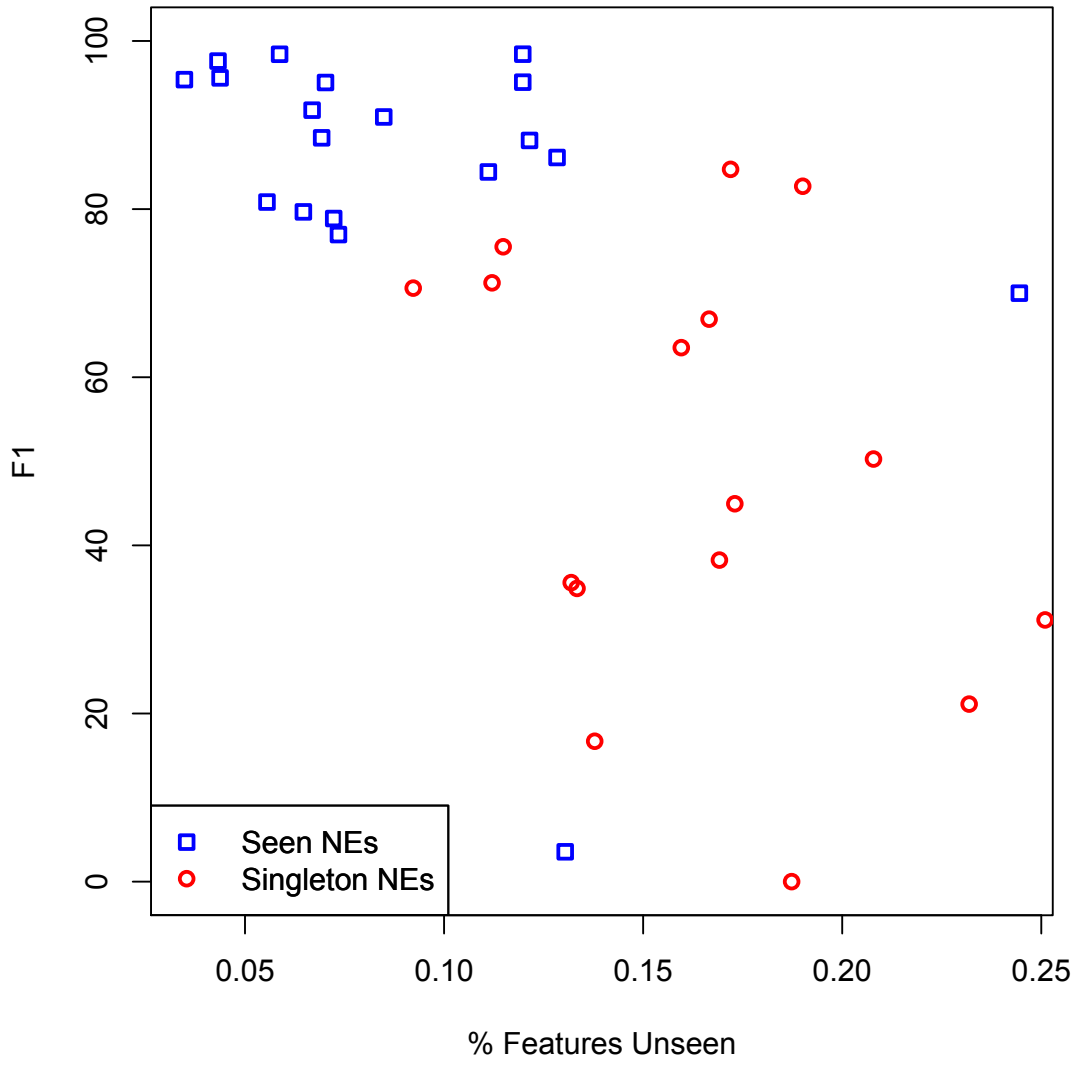

Figure 5.2: Percentage of unseen features and F1 with Stanford NER for seen and unseen NEs in different corpora

guidelines. In contrast, for the ConLL corpora, which belong to the same genre as Ontonotes NW, out-of-genre results are much lower than in-genre results. Memorisation performance on ConLL TestA and TestB with Ontonotes NW test suggest that this is partly due to the relatively low overlap in NEs between the two datasets. This is mostly likely due to the ConLL corpus having been collected in a different time period to the OntoNotes corpus, when other entities were popular in the news. In addition, there are differences in annotation guidelines between the two datasets.

The lowest F1 of 0 is achieved on ACE BN. An examination of that corpus reveals the reason

for this – the NEs contained in that corpus are all lower case, whereas those in OntoNotes NW have initial capital letters.

Corpora for which out-of-genre F1 is better than in-genre F1 for at least one of the NERC methods are: MUC7 Test, ACE WL, ACE UN, OntoNotes WB, OntoNotes TC and UMBC. Most of those corpora are very small, with combined training and testing parts containing fewer than 1,000 NEs (MUC7 Test, ACE UN, UMBC). In such cases, it appears beneficial to have a larger amount of training data, even if it is from a different genre and/or time period. The remaining 3 corpora contain Weblogs (ACE WL, ACE WB) and online discussions in the forum usenet (ACE UN). Those three are diverse corpora, as can be observed by the relatively low NEs/Unique NEs ratios (Table 5.3). However, NE/Unique NEs ratios are not an absolute predictor for better out-of-genre than in-genre performance: there are corpora with lower NEs/Unique NEs ratios than ACE WB which have better in-genre than out-of-genre performance.

To conclude, there are different types of genres which should be considered: is the corpus a subcorpus of the same corpus as the training corpus, does it belong to the same genre, is it collected in the same year, and was it created with similar annotation guidelines. Yet it is very difficult to explain high/low out-of-genre performance compared to in-genre performance with those factors.

A consistent trend is that, if out-of-genre memorisation is better in-genre memorisation, out-of-genre NERC performance with supervised learning is better than in-genre NERC performance with supervised learning too. It also reinforces what has been discussed in previous sections: an overlap in NEs is a very good predictor for NERC performance. This is a very useful conclusion for use cases in which a suitable training corpus for a new genre has to be identified. It can be time-consuming to engineer features or study and compare machine learning methods for different genres, while memorisation performance can be checked quickly.

Results on unseens for out-out-genre setting are in Table 5.11. What was reported in the last section about NERC performance being lower for unseen than for seen NEs is also true for the out-of-genre setting. The macro average over F1 for the in-genre setting is 76.74% for seen NEs vs. 53.76 for unseen NEs, whereas for the out-of-genre setting the F1 is 56.10% for seen NEs and 47.73% for unseen NEs. Corpora with a particularly big F1 difference (<= 20% averaged over all NERC methods) between seen and unseen NEs are ACE NW, ACE BC, ACE UN, OntoNotes BN and OntoNotes MZ. For some corpora, out-of-genre F1 (macro average over all methods) of unseen NEs is better than for seen NEs, these are ConLL Test A and B, MSM and Ritter.

5.5.4 RQ4: Memorisation, Context Diversity and NER performance

Having examined the impact of seen/unseen NEs on NERC performance in RQ2, the goal is now to establish the impact of seen features, i.e. features appearing in the test set that are observed also in the training set. While feature sparsity can help to explain low F1, it is not a good predictor of performance across methods: sparse features can be good if mixed with high-frequency ones. For instance, the results are better for Stanford NER (see Table 5.6), although the proportion of seen features is lower for that feature set than for CRFSuite. Also, some approaches such as SENNA use a small number of features and base their features almost entirely on the NEs and not on their context.

Subsequently, the goal is to measure F1 for unseen NEs and seen NEs, as in Section 5.5.2, but also observe the role of the proportion of seen features on the result. Therefore, the proportion of unseen features per unseen and seen proportions of different corpora is measured, an analysis of this with Stanford NER is shown in Figure 5.2. Each data point represents a corpus, the blue squares are data points for seen NEs and the red circles are data points for unseen NEs.

The figure shows that there is a clear negative correlation between F1 and percentage of unseen features, i.e. the lower the percentage of seen features, the higher the F1. Further, the percentage of seen features is higher for seen NEs. Note "seen NEs" and "seen features" cannot be separated clearly, as some of the features are extracted from the NE mention. This depends on the feature extraction method used.

For all aproaches the proportion of observed features for seen NEs is bigger than the proportion of observed features for unseen NEs, as it should be. However, within the seen and unseen testing instances, there is no clear trend indicating if having more observed features overall increases F1 performance. One trend that is observable is that the smaller the token/type ratio is (Table 5.2, the bigger the variance between the smallest and biggest n for each corpus, or in other words the smaller the token/type ratio is, the more diverse the features.

5.6 Conclusion

This chapter investigated reasons for poor NER performance on diverse genres. The goal is to study and compare NER performance on corpora from diverse domains and genres, including newswire, broadcast conversation, the Web and social media (see Table 5.1 for details.

The corpora are analysed with respect to their NE diversity, token/type diversity (Tables 5.3 and 5.2) and tag density. Corpora traditionally viewed as noisy such as Twitter corpora and Web corpora have a low NE/Unique ratio and token/type ratio, indicating they have a high repetition of NEs and tokens. Surprisingly this also applies to the ConLL corpus, which is most widely used corpus for NERC, indicating that it is well balanced in terms of stories.

The first research question is whether existing NER approaches generalise well over training data in diverse genres. This can be answered by comparing F1 performance across corpora and methods. All three systems, in particular, perform well on regular content such as newswire, but struggle on more diverse corpora like social media and Web text. However, results on size normalised corpora indicate that this effect may mainly be due to corpus size, suggesting that for more diverse genres such as Web corpora and social media corpora, manually adding training data would equally improve results. Although the experiments show a correlation between NERC performance and training corpus size, it is not an absolute predictor of F1.

The next research question investigates the impact of unseen NEs on NER performance. The F1 of unseen NEs is significantly lower than the F1 of seen NEs, which is a consistent trend across corpora. Moreover, test corpora with a high proportion of unseen NEs achieve a low performance. Out of the three approaches, SENNA is better at generalising than other approaches, probably because it makes use of word embeddings.

NE diversity explains part of the F1 differences of NER approaches between genres, but not all

of it – there are still significant F1 differences for seen NEs in different corpora. This means that NE diversity does not account for all of the difference of F1 between corpora of different genres.

The next research question explores how the difference in out-of-genre performance on unseen NEs and seen NEs compares to in-genre performance, the training corpus being the biggest corpus belonging to the newswire genre, Ontonotes NW. Experiments reveal that an overlap between NEs of the training and the testing corpus is a good predictor for high F1, as well as a high NE memorisation baseline. This is a very useful conclusion for use cases in which a suitable training corpus for a new genre or domain has to be identified. It can be time-consuming to engineer features or study and compare machine learning methods for different genres, while memorisation performance can be checked quickly. For some corpora, out-of-genre performance is better than in-genre performance. This is particularly for corpora containing Web and online discussion forums, as well as small corpora. This suggests that if only a small in-genre training corpus is available, applying an out-of-genre training corpus can lead to better results.

Finally, another factor that can impact NER performance is the proportion of seen features. Findings are that feature sparsity can help to explain F1, but it is not a good indicator for performance across methods: sparse features can be good if mixed with high-frequency ones. Further, there is a negative correlation between F1 and percentage of unseen features, the lower the percentage of unseen features, the higher the F1.

Overall, the experiments show that F1 is highest for regular corpora such as newswire corpora and lower for Web data and social media data, which is to a large degree due to the small size of corpora for diverse genres such as Web and social media. Unseen NEs pose a significant challenge and the proportion of unseen NEs is the best predictor for low F1. The proportion of seen features helps to explain differences in F1 across corpora, but not across methods. A very interesting finding is that out-of-genre performance can be better than in-genre performance if the available in-genre corpus is very small.

This leads to the following research questions which can still be investigated in future work:

What other factors influence NE performance, e.g. how much does NER performance differ for NEs of different lengths? What about the regularity of NEs (spelling, capitalisation)? It already helps if parts of NEs are seen, because features typically include stems of NE mentions etc as well. If NE training tests are very small, out of genre performance can be better than in-genre performance and could be improved even further using domain adaptation methods. At what size of domain-specific training data is the "turning point" at which in-domain performance is better than out-of-genre performance with domain adaptation or transfer learning methods?

What is the way forward for improving NERC performance for diverse genres, especially with limited resources? Is it to spend resources on creating more training data? How long would those resources be useful for since the unseen NE problem (*entity drift*) will increase over time? Is it to use domain adaptation methods or combine different NERC methods since they might be complementary to some degree?

What was learned in this chapter with respect to named entity recognition and classification for relation extraction from Web documents with distant supervision is that NERC performance depends on a variety of different factors: the availability of large training corpora, a high overlap

of NEs between training and testing corpora and suitable feature representations.

One possible future research avenue would be to test how the different research corpora investigated could be used and combined for a high relation extraction performance. This could involve research on domain adatation or transfer learning. Results suggest that out of genre corpora might even be better than in-genre corpora if the available in-genre corpora are small. However, for the distant supervision experiments, no real in-genre manually annotated NERC training corpus is available. Even if an NERC on, e.g. a combination of ACE and OntoNotes Weblog corpora were trained, this would still be different from the distant supervision Web data. It would also be possible to manually annotate some NERC data. However, the results above have shown that in-genre NERC performance is only high if the in-genre training corpus is large.

Therefore, a different method is investigated which is described in the next chapter. The method is based on the idea of using automatic relation labels created with distant supervision to train a relation extractor and a named entity classifier jointly in a way that the dependency between the two tasks is learnt. This is modelled using the imitation learning algorithm DAgger (Ross et al., 2011), which is a structured prediction method that can decompose tasks such as relation extraction into actions, such as named entity classification and relation extraction. Further, latent variables can be incorporated, which for the relation extraction task investigated in this thesis means that labels for NEs do not have to be known.

Chapter 6

Extracting Relations between Diverse Named Entities with Distant Supervision and Imitation Learning

6.1 Introduction

One of the main challenges for distant supervision for the Web, as experiments in Chapter 4 have shown, is to identify and assign types to arguments of relations, i.e. named entity recognition (NER) and named entity classification (NER) of arguments of relations. The core contribution of this chapter is a **novel method for NERC for Web-based distant supervision** based on joint learning of NEC and RE with imitation learning (also known as *inverse reinforcement learning*) which outperforms the state of the art[1]. In order to identify arguments of relations, named entity recognition and classification approaches, such as the Stanford NER tool, are typically applied by related work described in Section 2.4.4 and also in Chapter 4. However, as observed in those experiments, off-the-shelf NERC tools such as Stanford NER fail to recognise some NEs, especially those which are not persons, locations or organisations. This issue becomes more important as focus is shifting from using curated text collections such as Wikipedia to texts collected from the Web via search queries. Such Web-based distant supervision approaches can provide better coverage than distant supervision with static corpora (West et al., 2014). One of the reasons for NER mistakes could be the difference in domain or genre between training and testing data. For the setting reported in Chapter 4, the off-the-shelf pre-trained Stanford NER is used, which is trained with the ConLL 2003 task data belonging to the newswire genre, whereas the testing genre is Web data.

[1]The experiments in this chapter are based on a publication in the proceedings of the 20th Conference on Empirical Methods in Natural Language Processing (Augenstein et al., 2015b).

The previous section has explored reasons for NERC failure in diverse genres on existing gold standard corpora. The purpose of those experiments was to inform further research on NERC for distant supervision and determine if existing NERC gold standards for Web data might be useful for such research.

Considering those experiments, one possible (and obvious) way of improving NERC for distant supervision could be to research methods for supervised NERC for Web-based distant supervision, based on a combination of existing NERC gold standard corpora. Such methods could make use of domain adaptation or transfer learning to adapt to new genres, or e.g. drift compensation methods if the training and testing corpora are from different years (Derczynski et al., 2015a). However, as experiments in Chapter 5 have shown, it might not be straightforward to find suitable NERC training data, and transfer learning methods might only bring small improvements. Moreover, such a method would always depend on gold standard NERC corpora, meaning results would change with new relation extraction testing data or new relation types and genres.

Therefore, a different approach is researched in this chapter, which does not rely on any NERC gold standard data, does not require any manual effort and is easily portable to new genres. A further benefit of the proposed approach is error reduction by decomposing the task of RE into several subtasks, then jointly learning classifiers for those tasks. Even if a suitable NERC were available, applying it as a preprocessing stage for RE leads to errors made by the NERC stage being propagated to the RE stage. Solutions to this problem have been researched for supervised relation extraction approaches. Some works in the recent years have focused on solving the problem by proposing joint inference frameworks so that the two tasks can learn to enhance one another. Early strategies include re-ranking (Ji and Grishman, 2005), integer linear programming (Roth and Yih, 2004, 2007; Yang and Cardie, 2013) and card-pyramid parsing (Kate and Mooney, 2010). While those models all exploit cross-component interactions, they are based on models separately trained for each subtask. Approaches which jointly model subtasks using a single model are based on probabilistic graphical models (Domingos et al., 2008; Yu and Lam, 2010; Singh et al., 2013) or incremental beam search (Li and Ji, 2014). Such approaches could also be beneficial for distant supervision, but since they require NERC labels and such labels are not available for distantly supervised RE, modelling a joint approach for distantly supervised NERC and RE is not as straightforward.

The approach proposed in this chapter is based on the imitation learning algorithm DAgger (Ross et al., 2011), which is used to learn the NEC component jointly with relation extraction (*RE*), without requiring explicitly labeled data for NERC. Instead, a training signal for NEC is obtained by assessing the predictions of the relation extraction component. Named entities are identified in a NER step which makes use of Web-based and part of speech heuristics.

To summarise, the following contributions are made in this chapter:

1. **A novel method for NERC for distant supervision:** A named entity classifier and a relation extractor are trained jointly for Web-based distant supervision, after identifying entity pairs using Web-based and part-of-speech-based heuristics. The method does not rely on hand-labeled training data and is applicable to any domain, which is shown in an evaluation on 18 different relations.

2. **Empirical comparison to state of the art:** Different methods are compared for this purpose: (1) imitation learning is used to train separate classifiers for NEC and RE jointly; (2) NEC features and RE features are aggregated and a one-stage classification model is trained; (3) a one-stage classification model with only RE features is trained; (4) NEs are classified with two supervised off-the-shelf NEC systems (Stanford NER and FIGER) and the NE types are used as features in RE to achieve a soft NE type constraint.
3. **Features:** The effects of using different NEC and RE features are explored, including Web features such as links and lists on Web pages, and it is shown that Web-based features improve average precision by 7 points. Further findings are that high-precision, but low-frequency features perform better than low-precision and high-frequency features.
4. **Demonstrated improvement over state of the art:** The experiments show that joint learning of NEC and RE with imitation learning outperforms one-stage classification models by 4 points in average precision, and models based on Stanford NER and FIGER by 19 and 10 points respectively.
5. **Corpus:** The Web-based corpus used in experiments in this chapter is made publicly available. The corpus is annotated and is made available in the same curated format as Angeli et al. (2014b)'s distant supervision data for easy reuse. Although the same collected Web pages are used for this corpus as for the experiments in Chapter 4, the resulting corpus is different and of higher quality, as the result of the researched relation candidate identification methods.

Early experiments on imitation learning for relation extraction for the architecture domain have already been performed by Vlachos and Clark (2014b). The work documented in this chapter is novel compared to that early work in the following sense:

- Candidates for joint NEC and RE are identified using a number of Web-based and part-of-speech-based named entity recognition heuristics depending on the coarse NE type (e.g. person, location, organisation) of the object of the relation. No such experiments are performed by Vlachos and Clark (2014b)
- The methods proposed in this chapter are empirically compared to the state of the art, whereas Vlachos and Clark (2014b) only use internal baselines
- Feature extraction methods based on Web markup are proposed. Extensive experiments on using different NE and relation features, including Web-based features, are performed and evaluated.
- Experiments are performed on a large Web corpus spanning different domains, whereas Vlachos and Clark (2014b) only evaluate their approach for two relations for the architecture domain.

6.2 Background on Imitation Learning

Imitation learning is also referred to as search-based structured prediction (Daumé et al., 2009), learning to search (Chang et al., 2015b) or inverse reinforcement learning (Abbeel and Ng, 2004). As such, its roots are in reinforcement learning (Sutton and Barto, 1998).

The idea of reinforcement learning is that an agent can learn, by demonstration from a teacher, which actions to take in a particular state to maximise a reward (or minimise cost) towards a certain goal. Crucially, the actions are not known, but the agent must discover which actions lead to the lowest cost by taking them. Further, an action may not only affect the current cost but also costs for future states, which is known as *delayed reward*. Reinforcement learning methods exploit sequences of action known to lead to high reward and then progressively learn new action sequences by exploration. For example, in the natural language dialogue system for restaurant recommendation by (Rieser and Lemon, 2010), possible actions are: to recommend one specific restaurant, to compare a number of restaurants in detail, or to give a brief summary of how the restaurants differ, or a mixture of these.

In contrast, supervised learning is learning by studying training examples. This assumes training examples representative for different action sequences are available. Reinforcement learning, on the other hand, only assumes that training signal is available in the form of a reward function, and that the agent can learn by itself through exploration. Further, supervised learning often considers subproblems without considering how they might be useful. In the area of NLP, pre-processing components such as part of speech taggers or named entity recognisers are often learned as stand-alone components without being trained directly for a larger application or goal. Reinforcement learning considers concrete larger goals, e.g. learning to play complicated games such as Atari (Guo et al., 2014) or Go (Silver et al., 2016), or learning a dialogue system (Rieser and Lemon, 2010).

The central components of a reinforcement model are the following (Sutton and Barto, 1998):

- *Agent*: An agent learns how to achieve a goal in an environment by interacting with it.
- *Environment*: An environment is what an agent interacts with, i.e. everything outside the agent which is part of the learning problem.
- *Policy*: A policy defines the behaviour of an agent and maps states to actions. This can be based on lookup tables or can be stochastic. For instance, in the dialogue system by (Rieser and Lemon, 2010), one aspect defined in their policy is that if in the current state, the action chosen was to compare different restaurants, in the next state the possible actions are to recommend a specific restaurant or to end the dialogue.
- *Reward function*: A reward function defines the goal of the learning problem. It maps state-action pairs to numbers indicating how desirable it is to take actions in a certain state. The reinforcement learning agent's objective is to achieve a high total reward. Reward is the opposite of the concept of cost, i.e. a high reward is similar to a low cost. In reinforcement learning, the reward function is part of the input to the learning problem, but does not have to be defined manually, e.g. Rieser and Lemon (2010) extract it automatically as part of their preprocessing.
- *Value function*: Values are predictions of rewards, which indicate which actions are desirable in the long term. The value function for a certain state is computed by taking into account

the reward of states that are likely to follow it. This means a value function for a certain state might be low although the reward for a certain state is high or vice versa.

Imitation learning borrows some of those notions from reinforcement learning. The main difference between imitation and reinforcement learning concerns the reward function (Russell (1998); Ziebart et al. (2008); Daumé et al. (2009)). In reinforcement learning, the reward function is known and the agent uses it to learn to behave in an environment, i.e. the goal of learning is to maximise the total reward. In imitation learning, the situation is reversed: the reward function is unknown and the goal of learning is to recover it. During the imitation learning process, reward weights are found to imitate the behaviour of a demonstrator.

Imitation learning algorithms such as SEARN (Daumé et al., 2009) and DAGGER (Ross et al., 2011) are algorithms for solving structured prediction problems. Whereas prediction is the task of learning a function that maps input to an output such as -1 or 1, in structured prediction, the output has a more complicated structure. The output can e.g. be a sequence of part of speech labels given an input sentence, a sequence of word in a target language given a sequence of words in a source language, or a sequence of information extraction stages (named entity classification, relation extraction), as discussed further in this chapter.

Imitation learning algorithms are able to learn structured prediction models without the need to decompose loss functions, for arbitrary features and with imperfect data, e.g. missing data. Examples of non-decomposable loss functions commonly used in NLP are F1 or BLEU (Papineni et al., 2002).

Imitation learning algorithms for structured prediction decompose the prediction task into a sequence of actions for which simple classifiers can be learned, using cost sensitive classification (CSC) learning. Classifiers are trained to take into account the effect of their predictions on the whole sequence by assessing their effect using a (possibly non-decomposable) loss function on the complete structure predicted. The dependencies between the actions are learnt via appropriate generation of training examples.

Sample applications of imitation learning include biomedical event extraction (Vlachos and Craven, 2011), dynamic feature selection (He et al., 2013), machine translation (Grissom et al., 2014) and dependency parsing (Chang et al., 2015a).

The components of an imitation learning approach are (Daumé et al., 2009):

- *A search space*: A search space determines the set of possible actions. This can be the set of part of speech tags, a set of relation types, or in a simple binary classification case, *true* or $false$.

- *A cost sensitive classification learning algorithm*: A classification algorithm which can be trained using cost sensitive classification data. The idea of cost sensitive classification is that some mistakes should cost more than other mistakes. Cost sensitivity is optional, in a simple case, as also considered in this chapter, the cost is either 0 (for correct predictions) or 1 (for incorrect predictions).

- *Labelled structured prediction training data*: Training data for cost-sensitive classification. Each training example consists of a sequence of states. At each state, several actions are

possible, as determined by the search space. Training labels must be provided for the final state, but can otherwise be incomplete.

- *An expert policy*: A policy similar to one used in reforcement learning, which maps states to actions. The expert policy should achieve low loss on the training data, but does not need to be perfect. The expert policy, in combination with the labelled input data, is used to create cost-sensitive classification examples. The goal of this is to explore the search space and, finally, learn a trained policy to generalise on new, unseen data.
- *A trained policy*: The trained policy is learned based on the cost sensitive classification examples and the cost sensitive classification algorithm. Unlike the expert policy, it can generalise to unseen data.

The attraction of imitation learning for the distantly supervised relation extraction problem is that the components (here: named entity classification, relation extraction) are learnt jointly. Further, because imitation learning can cope with missing data, only labels for the output are required (relation extraction), but not intermediate steps (named entity classification). Note that named entities still have to be recognised. This is solved here by using simple Web-based and part-of-speech-based heuristics which identify relation candidates with high recall. At test time, the two learned models (named entity classification, relation extraction) are applied in sequence to the relation candidates. Only if both models predict a positive label for the relation in question and the testing instance is an overall positive prediction made.

Experiments described in this chapter were performed with the DAgger (Ross et al., 2011) algorithm. Section 6.4.1 explains the algorithm and how it is applied to relation extraction in more detail.

6.3 Approach Overview

The input to the approach is a KB which contains entities and is partly populated with relations, the task is to complete the knowledge base. As an example, consider a KB about musical artists and their albums, which contains names of musical artists, and albums for some of them. The task is then to find albums for the remaining musical artists. Queries are automatically formulated containing the class of the subject C, the subject s and the object o, e.g. "Musical Artist album 'The Beatles"' and Web pages are obtained using a search engine. For each sentence on the Web pages retrieved which contains s, all candidates for C are identified using NER heuristics (Section 6.4.2). Next, the distant supervision assumption is applied to all such sentences containing s (e.g. "Michael Jackson") and a candidate for that relation (e.g. "Music & Me"). If the candidate is an example of a relation according to the KB, it is used as a positive example, and if not, as a negative example. The examples are then used to train a model to recognise if the candidate is of the right type for the relation (NEC) and if it is of the correct relation (RE). The model is applied to the sentences of all the incomplete entries in the KB. Since different sentences could predict different answers to the query, all predictions are combined for the final answer.

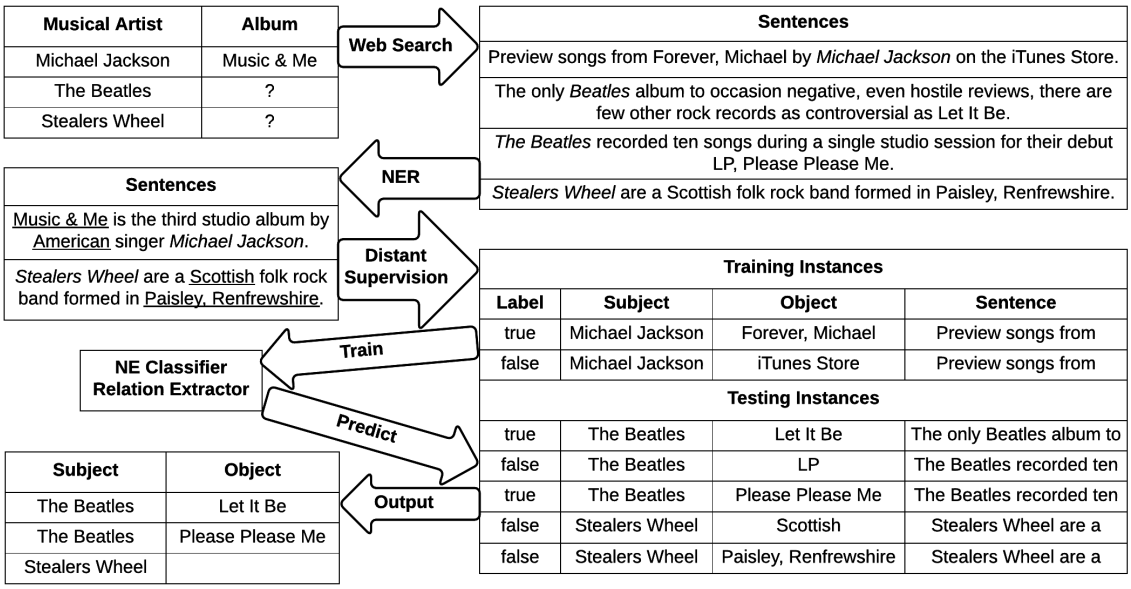

Figure 6.1: Overview of approach

The core approach described above is the same as in Chapter 4 with two exceptions: 1) a modification for joint modelling of NERC and relation extraction, meaning features are extracted for NEC and RE, and classifiers for both of those tasks are learned; 2) instead of multi-class classification as in Section 4, binary classification is performed at both stages. One binary classifier for the NEC and RE stages for each relation is trained on the training data for each relation and then at test time, those two (NEC and RE) classifiers are applied to the testing data for that relation. This is so that it is easier for different candidate identification heuristics for each relation to be evaluated, but it could easily be changed to the same multi-class setting as in Section 4.

6.4 Named Entity Recognition and Relation Extraction

The input to the learning task is a collection of training examples for a specific relation. The examples are sentences containing the subject of the relation and one further NE identified using simple heuristics. The examples are labeled as true (relation is contained in knowledge base) or as false (relation is not contained in the knowledge base).

The task is modelled in two binary classification stages: named entity classification (NEC) and relation extraction (RE). Existing approaches assume that named entity recognition and classification is done as part of the pre-processing. However, this is not possible in domains for which NE classifiers are not readily available. To ameliorate this issue, existing approaches — e.g Mintz et al. (2009) — perform NEC to provide additional features for relation extraction. Two such baselines with off-the-shelf NECs are used here, for which the NE labels are added to the relation features. The first baseline (**Stanf**) is with the Stanford NER 7-class (Time,

Location, Organization, Person, Money, Percent and Date) model, the second (**FIGER**) is with the fine-grained FIGER (Ling and Weld, 2012).

An alternative approach is to simply add NEC features to relation extraction features, which is called **one-stage model (OS)** here. NEC features are typically morphological features extracted from the NE mention and features to model its context, whereas relation features typically model the path between the subject and object of the relation. While NEC features may be useful to determine if the NE has the correct type for the relation, such features are usually less sparse and also not directly related to the relation extraction task. Consider the following sentence, containing an example of the relation director:

"One of director <o>Steven Spielberg</o>'s greatest heroes was <o>Alfred Hitchcock</o>, the mastermind behind <s>Psycho</s>.

This sentence contains two relation candidates, "Steven Spielberg" and "Alfred Hitchcock", between which the decision for the final prediction has to be made. Both of the candidates are directors, but only one of them is the director of "Psycho". Because the context around "Steven Spielberg" is stronger (preceded by "director"), NEC features alone are more likely to indicate that as the correct candidate and also likely to overpower relation features for the final prediction, as the latter tend to be sparser.

Ideally, two models would be trained, one for NEC and one for RE, which would be applied in sequence. If the NEC stage concludes that the candidate is of the correct type for the relation, the RE stage determines whether the relation between the two entities is expressed. If the NEC stage concludes that the entity is not of the correct type, then the RE stage is not reached. However, distant supervision only provides positive labels for NEC, since if a sentence is labeled as false it is unknown if that is due to the candidate not being of the correct type, or the relation not being true for the two entities. To overcome this, models for the two stages, NEC and RE, are learned jointly using the imitation learning algorithm DAgger (Ross et al., 2011), as described in the next section.

6.4.1 Imitation Learning for Relation Extraction

The ability to learn by assessing only the final prediction and not the intermediate steps is very useful in the face of missing labels, such as in the case of missing labels for NEC. The imitation learning problem consist of two stages (similar to the "states" in standard reinforcement learning terminology): an NEC stage and an RE stage. Possible actions for both stages are "true" or "false". Action sequences then consist of one NEC action ("true" or "false") and possibly one RE action, dependent on whether the NEC action is "true", i.e. the entity is of the appropriate type for the relation.

For each training instance, supervision for the NEC stage is obtained by taking both options for this stage, "true" or "false", obtaining the prediction from the RE stage in the former case and then comparing the outcomes against the label obtained from distant supervision. Thus the NEC stage is learned so that it enhances the performance of RE. In parallel, the RE stage is learned using only instances that actually reach this stage. The process is iterated so that the models learned adjust to each other.

Algorithm 2 Imitation Learning with DAgger (Ross et al., 2011)

Input: training instances $\mathcal{S}$, expert policy π^*, loss function ℓ, learning rate β, number iterations N, CSC learner CSCL

Output: learned policy π_N

Initialise CSC examples $E = \emptyset$
Initialise $\pi_0 = \pi^*$
for $i = 1$ **to** N **do**
  $p = (1 - \beta)^{i-1}$
  current policy $\pi_i = p\pi^* + (1 - p)\pi_{i-1}$
  for s **in** $\mathcal{S}$ **do**
    predict $\pi_i(s) = \hat{y}_{1:T}$
    for $\hat{y}_t$ **in** $\pi_i(s)$ **do**
      extract features ϕ_t from s and $\hat{y}_1 \dots \hat{y}_{t-1}$
      for each possible action y_t^j **do**
        Predict $y'_{t+1:T} = \pi_{i-1}(s; \hat{y}_{1:t-1}, y_t^j)$
        Assess $c_t^j = l(\hat{y}_{1:t-1}, y_t^j, y'_{t+1:T})$
      end for
      Add (ϕ_t, c_t to E)
    end for
  end for
  learn a new policy $\pi_i = CSCL(E)$
  decrease p
end for
Return π_N

Algorithm 2 contains a detailed description of the DAgger imitation learning algorithm, based on Ross et al. (2011); Vlachos (2012), applied to structured prediction for relation extraction.

The algorithm requires a set of training instances S, a loss function l that compares output predictions against the gold standard, an expert policy π^*, a number iterations N, a learning rate β and a CSC learner (CSCL). The structured output prediction for an instance s consists of a sequence of T actions. The expert policy returns the optimal action $\hat{y}_t$ for each instance based on the RE data labelled using distant supervision. The CSC learner remembers misclassification costs, so that some mistakes are more expensive than other mistakes. In this case, since losses are either 0 or 1, CSC is similar to normal classification algorithms. The CSCL learns dependencies between actions, in this case a sequence of the two actions NEC and RE. The RE stage is only reached if the NEC stage is positive. For CSCL, the passive aggressive algorithm (PA, Crammer et al. (2006)) is used. The output is a learned policy π_N which can generalise to unseen data.

For each iteration $1 \ldots N$ the probability p of using the expert policy π^* in the current policy π_i is set. The probability p determines how likely it is that the expert policy π^*, which is derived from the labelled training data, is chosen as the current policy, whereas the probability $1-p$ is how likely it is that the policy learned in the previous iteration π_{i-1} is chosen (Line 5). An exception to this is the first iteration, in which the expert policy is chosen by default. The probability p depends on the learning rate β (Line 4), a value between 0 and 1, which determines how fast PA moves away from the expert policy.

The current policy π_i is then used to predict the instances (Line 7). Lines 8-15 shows how CSC examples are then generated for each instances s and each action $\hat{y}_t$ (NEC, RE). First, features are extracted for each of the actions NEC and RE features (Line 9), described in more detail in Sections 6.4.2 and 6.4.3. The cost for each action is then estimated by predicting the remaining actions $y'_{t+1:T}$ (Line 11), for this it is assumed that the action was taken, similar to a look-ahead. The cost for both the NEC stage and the RE stage is -1 if the choice of action leads to the RE stage predicting the correct answer and 1 otherwise (Line 12). The current CSC training example is then defined as the features for each instance and cost for each action (Line 14) and is combined with CSC training examples from previous iterations to learn a new policy π_i (Line 17). After N iterations, the policy learned in the last iteration, π_N, is returned.

To give a more concrete idea of how the models are trained and applied, consider the example introduced in the previous section of trying to extract albums for a given set of musical artists. Training instances for IL are generated from the crawled Web corpus. 1/3 of the training instances are set aside initially for tuning decision thresholds. Candidates for the relation "album" are extracted using the candidate identification strategies explained in Section 6.4.2. These candidates are in the form illustrated in Figure 6.1, i.e. the training data for the relation "album" might consist of the following two training instances, S:

Ex1: label: true ; subject: Michael Jackson ; object: Forever, Michael ; Sentence: Preview songs (...)
Ex2: label: false ; subject: Michael Jackson ; object: iTunes Store ; Sentence: Preview songs (...)

The learning rate β used for the experiments documented in this chapter is 0.25, the number of iterations N to 12 and for cost sensitive classification learning (CSCL), the classifier PA Crammer

et al. (2006)) is used. Note that CSCL involves training two classifiers, one for the NEC stage and one for the RE stage. Before the start of training, the CSC training examples are initialised to an empty set (Line 1). Those training examples are generated by the imitation learning algorithm during training. Next, the initial policy is set to be the same as the expert policy for the first iteration. In the first iteration, p, the probability of choosing the expert policy is 1, afterwards it decreases.

In Line 7, the policy from the previous iteration is used to predict labels for each training instance. In the first iteration, since no trained policy is available, the policy is entirely based on the labelled training data. It predicts the relation labels specified as part of the training data, so for the example above, "true" (Ex1) and "false" (Ex2). The policy makes structured output predictions, consisting of 1 (NEC) or possibly 2 actions (NEC, RE). The first action $\hat{y}_1$ is an NEC action ("true" or "false") and the second $\hat{y}_2$ is a RE action ("true" or "false"). The RE stage is only reached if the action at the NEC stage is "true". Assuming a policy which makes correct predictions for those two training instances, the policy would predict the action sequences $< true, true >$ for Ex1 and $< false >$ for Ex2. However, recall that a negative label for RE does not mean that the NE type is incorrect. The object could have the correct NE type and still be wrong for the relation, e.g. it could be the name of an album for a different musical artist. In practice, the correct NEC labels are not available, so the expert policy has to be defined to take that into account. For the experiments documented in this chapter, the expert policy is defined so that the optimal action for NEC is always "true". Note that the expert policy is different from the final learned policy, i.e. the final NEC model will not predict "true" for all instances. Since the models for NEC and RE are learned to enhance one another, this eventually, after a number of iterations N, leads to a learned policy π_N with a permissive NEC stage and a stricter RE stage. Other options for defining the expert policy for NEC would be to always predict "false" if the RE label is "false", or to randomise the selection of "true" or "false" for the expert policy NEC action. Empirically, defining the expert policy for NEC so that it always predicts "true" only leads to small improvements over defining it to predict "false" or with a randomised selection.

Lines 8 to 15 detail how CSC examples are generated. First (Line 9), features for the NEC action (ϕ_1) and features for the RE action (ϕ_2) are extracted for each training instance, as detailed in Sections 6.4.2 and 6.4.3, e.g. for NEC, the object occurrence (Ex1: Forever, Michael; Ex2: iTunes Store) and for RE, the dependency path between Michael Jackson and the object occurrence is extracted.

Next, for each action y_t^j (Line 10), the remaining actions are predicted (Line 11). The predicted label of the final action y_2^j is compared to the labelled training data to calculate the cost of each action y_t^j and estimate the overall loss using the loss function l (Line 12). The cost of an action y_t^j is 1 if it leads to the last action of the action sequence y_2^j (RE stage) predicting an incorrect label, otherwise it is -1. For instance, for Ex1, if the NEC action is "false", the RE label "false" is predicted, so the cost for both NEC and RE is 1. For Ex2, if the NEC action is "true" and the RE action is predicted as "false", the cost for both NEC and RE is -1 since the RE prediction "false" is correct for Ex2.

The features and costs per training instance for each action at each timestep are saved as one

CSC example (Line 14). Note that CSC examples are generated using π_{i-1}, the policy from the previous iteration, for different action sequences for NEC and RE. If the NEC action is "false", no further action is explored and the predicted label is "false". However if the NEC action is "true", then the explored RE actions depend on the RE prediction by the policy π_{i-1}. Recall that the possible action sequences to be explored are $< true, true >$, $< true, false >$ or $< false >$. Following generating CSC examples, the CSC examples from the current iteration are added to the ones from the previous iterations (Line 14) and a new policy π_i is trained on them. Policy training involves learning one binary classifier for NEC and RE each on the respective NEC or RE feature sets. Note that the feature sets are features extracted in the current iteration and features extracted in all previous iterations. After N iterations, the resulting learned policy π_N is returned and applied to the held out 1/3 of training instances. The output is a score for each training instance and stage, e.g.

Ex1: NEC: 0.629 ; RE: 0.735
Ex2: NEC: -0.339 ; RE: 0.217

For each stage (NEC and RE), there is a decision threshold for deciding between the labels "true" and "false", the default is 0. New thresholds for both stages are picked based on predictions on the predictions on the development data, after which a new policy is trained on the full training set and applied to all instances in the test set. For each relation, relation candidates are again identified using the relation candidate identification strategies, which differ depending on the relation, as documented in Section 6.4.2. Afterwards, the two trained binary PA models (NEC and RE) which form the trained policy are applied to the testing data in sequence. As also during training, if the NEC classifier or the RE classifier predict "false" for a testing instance, this is returned as the final prediction it. Otherwise, if the RE classifier predicts "true", the final prediction for the testing instance is "true".

6.4.2 Relation Candidate Identification

To assign types to NEs (NEC) and extract relations among NEs (RE), as described earlier in this section, boundaries of those NEs have to be detected first. Most distantly supervised approaches use supervised NER systems for this, which, especially for relations involving MISC NEs, achieve a low recall. High recall for NE identification is more important than high precision, since precision errors can be dealt with by the NEC stage. For a relation candidate identification stage with higher recall, POS-based heuristics for detecting NEs[2] and HTML markup are utilised instead. The following POS heuristics are used:

- **Noun phrases:** Sequences of tags which start with *N*. Those include singular and plural nouns (*NN* and *NNS*), as well as singular and plural proper nouns (*NNP*, *NNPS*). This heuristic takes into account that the POS tagger does not always recognise proper nouns correctly and sometimes tags them as nouns instead.

[2]The Stanford POS tagger uses Penn Treebank POS tags, see http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html for a list of tags

- **Capitalised phrases:** Phrases with at least initial capital letters for each word (i.e. Dagger, DAgger, DAGGER would all qualify). Those can be distinct from noun phrases, e.g. some album titles are capitalised phrases ("Whatever People Say I Am, That's What I'm Not" by the Arctic Monkeys)

Further, words which contain the following HTML markup are considered as relation candidates:

- **Phrases from HTML markup:** All sequences of words marked as: `<ahref>` (links), `<li>` (list elements), `<h1>` or `<h2>` or `<h3>` (headers and subheaders, i.e. titles), `<strong>` or `<b>` (bold), `<em>` (emphasised), `<i>` (italics).

Those three different relation candidate identification strategies explained above are then applied depending on the coarse NE types of objects of relations as defined in the KB (Table 6.3).

- **PER**: All capitalised noun phrases. A maximum of two words are allowed to be surrounded by quotes to capture alternative first names, e.g. "Jerome David 'J. D.' Salinger".
- **LOC**: All capitalised noun phrases.
- **ORG**: All capitalised phrases and phrases from HTML markup. The latter is to capture ORG names for which not all words of the phrase start with capital letters, e.g. the school "Woodrow Wilson School of Public and International Affairs" or the record label "Sympathy for the Record Industry".
- **MISC**: As for ORG, all capitalised phrases and phrases from HTML markup are used. MISC NEs are even more varied than ORG NEs and it is difficult to find the right balance between recognising most of them and generating unecessary candidates since many MISC NEs are mixed case. By using the HTML markup candidate identification strategy, some of them can be identified without generating too many unecessary candidates.

To assess how useful these strategies are, 30 instances of each Freebase class per coarse NE type of the object are randomly sampled and all sentences which contain the subject of the relation examined manually. Precision is measured, i.e. how many of the relation candidates are appropriate for the relation, as well as recall to compare the relation candidate identification strategies described above against the identification of candidates by Stanford NER (ignoring the NE label). As shown in Table 6.1, while supervised identification of NE labels achieves a higher precision for all NE types, the recall is higher for all NE types using POS-based heuristics. The simple heuristics are especially helpful for MISC NEs, for which recall is twice as high compared to Stanford NER and precision only marginally higher. If the NE label were used to enforce hard constraints, recall would be reduced even further: 88% of all PER entities are correctly identified as persons, compared to 58% for locations and 87% for organisations. MISC NE are identified as PER (45%), LOC (40%) or ORG (15%). Overall, precision is not as important for candidate identification as recall, since choosing correct entities among the candidates can be dealt with in a NEC stage.

NEC features

For the one-stage and imitation learning model, the following **Web features** based on HTML markup are used, both as local features if the entity mention contains the markup, and as global

NE type	Model	R	P	F1
PER	heuristic	0.976	0.1287	0.227
PER	Stanford	0.774	0.1781	0.29
LOC	heuristic	0.963	0.1176	0.21
LOC	Stanford	0.889	0.1611	0.273
ORG	heuristic	0.95	0.0265	0.0516
ORG	Stanford	0.8	0.0505	0.095
MISC	heuristic	0.854	0.0496	0.0938
MISC	Stanford	0.427	0.053	0.0943

Table 6.1: Results for POS-based candidate identification strategies compared to Stanford NER

features if a mention somewhere else in the document with the same lexicalisation contains that markup:

- is link (`<ahref>`)
- is list element (`<li>`)
- is header or subheader (`<h1>` or `<h2>` or `<h3>`)
- is bold (`<strong>` or `<b>`)
- is emphasised (`<em>`)
- is italics (`<i>`)
- is title (`<title>`)
- is contained in title (`<title>`)

In addition, the following NEC features are extracted, based on Nadeau and Sekine (2007) and Hoffmann et al. (2011):
Word features (mentfeats):

- Object occurrence
- Sequence and BOW of occurrence
- Sequence and bag of POS of occurrence
- Number of words, characters and digits of object
- Ends with period, is roman number, contains apostrophe, hyphen, ampersand, possessive
- Digit and capitalisation pattern

Context features, as 1-grams (1cont) and 2-grams, 2 words to left and right of occurrence (2cont): BOW, sequence, bag of POS, POS sequence.

6.4.3 RE Features

The following features are used for RE, based on Hoffmann et al. (2011) and Mintz et al. (2009):

Person	
Musical Artist : album	Politician : birthplace
Musical Artist : record label	Politician : educational institution
Musical Artist : track	Politician : spouse
Organisation	
Business : employees	Education : mascot
Business : founders	Education : city
Mixed	
Film : director	Book : author
Film : producer	Book : characters
Film : actor	
Film : character	
Location	
River : origin	
River : mouth	

Table 6.2: Freebase classes and properties/relations used

- 1cont and 2cont features
- Flag indicating which entity came first in sentence
- Sequence of POS tags and bag of words (BOW) between the subject and the object occurrence

Parsing features as full sequences (parse):

- Dependency path between subject and object, POS tags of words on that path
- Lemmas on dependency path, same with NNP and CD tokens substituted by POS tags

6.4.4 Supervised NEC Features for RE

For the baselines with off-the-shelf NECs, sentences are preprocessed with the two NEC systems Stanford NER and FIGER. NE labels are then used in addition to the RE features listed in Section 6.4.3. For the Stanf baseline, Stanford NER 7-class labels are added as RE features. Those are: Time, Location, Organization, Person, Money, Percent, Date. FIGER classifies NEs according to 112 types, most of which are subtypes of Person, Organization, Location, Product, Art, Event and Building. Some of those types are relation types used in the evaluation of experiments documented in this chapter (see Table 6.3 for relation types): educational institution, city, director, actor and author. Since FIGER performs multi-label classification, it annotates some of the relation candidates with more than one NE label. In that case, all NE labels returned are added as features, though more experiments on how best to integrate multiple NE labels as features could be performed, as shown by Liu et al. (2014).

6.5 Evaluation

Musical Artist		**Politician**	
Relation type	**NE type**	**Relation type**	**NE type**
album	MISC	birthplace	LOC
record label	ORG	educational institution	ORG
track	MISC	spouse	PER
Business		**Educational Institution**	
Relation type	**NE type**	**Relation type**	**NE type**
employees	PER	mascot	MISC
founders	PER	city	LOC
Film		**Book**	
Relation type	**NE type**	**Relation type**	**NE type**
director	PER	author	PER
producer	PER	characters	MISC
actor	PER		
character	MISC		
River			
Relation type	**NE type**		
origin	LOC		
mouth	LOC		

Table 6.3: Relation types and corresponding coarse NE types

6.5.1 Corpus

To create a corpus[3] for Web RE, seven Freebase classes and two to four of their relations are selected (Table 6.2). The selected classes are subclasses of PER (Musical Artist, Politician), LOC (River), ORG (Business (Operation)), Education(al Institution)) or MISC (Film, Book), as can be seen in Table 6.3. For each entity, at most 10 Web pages were retrieved via the Google Search API using the search pattern "'*subject_entity*" *class_name relation_name*', e.g. "'The Beatles" Musical Artist Origin'.

This corpus is a subset of the corpus used in Chapter 4, see Section 4.3.1 for a more detailed description. The discarded relations are those which have a small set of possible objects, such as "Education: colors". It could be argued that those are easy to extract using e.g. simple gazetteer-based extraction methods and that therefore, more complicated methods such as the one proposed in this chapter are not necessary.

[3]The resources for experiments documented in this chapter are available online via http://tinyurl.com/o8ykn4y

Model	R-top	P-top	F1-top	R-all	P-all	P-avg
RelOnly	0.1943	0.404	0.255	0.223	0.309	0.373
Stanf	0.233	0.436	0.304	0.268	0.329	0.398
FIGER	0.228	0.497	0.298	0.251	0.413	0.483
OS	**0.269**	0.58	**0.356**	**0.288**	0.486	0.552
IL	0.246	**0.600**	0.329	0.271	**0.521**	**0.588**

Table 6.4: Results for best model for each relation, macro average over all relations. Metrics reported are first best precision (P-top), first best recall (R-top), first best F1 (F1-top), all precision (P-all), all recall (P-all), and all average precision (P-avg)(Manning et al., 2008). The number of all results for computing recall is the number of all relation tuples in the KB.

6.5.2 Models and Metrics

The following models are evaluated: imitation learning (**IL**) as described in Section 6.4.1, a one-stage model (**OS**), a one-stage model with relation features only (**RelOnly**), and using Stanford (**Stanf**) and FIGER (**FIGER**) NE labels as features (Section 6.4). For all models, linear classifiers learned with passive-aggressive updates are used. For imitation learning, the learning algorithm DAgger is used, which requires two parameters: the learning rate, i.e. how quickly the learning algorithm moves away from the expert policy, and the number of iterations. The best learning rate for this particular prediction task was empirically determined to be 0.25 and the best number of iterations 12.

The output of the models is a score for each relation example and stage, i.e. for the one-stage model, the output is one score and for the imitation learning model, there is a score each for the NEC stage and the RE stage. The default for deciding whether the relation label should be true or false depends on stage thresholds, which are 0 by default. Instead of using the default thresholds, thresholds are automatically picked for all models on 1/3 of the training set, which is set aside as a development set, after which models are retrained on the whole training set and used to predict relations based on the learnt thresholds.

The following metrics are used: first best precision (P-top), first best recall (R-top), first best F1 (F1-top), all precision (P-all), all recall (P-all), and all average precision (P-avg). For top, only the top-ranked answer is considered, whereas for all all answers are returned until either the correct one is found or they are exhausted. Finally, in the all mode precision is evaluated at all recall points by varying the thresholds used in the respective classifiers and average precision (P-avg) Manning et al. (2008) is reported. Merely reporting precision and recall ignores that those two measures are dependent on one another, i.e. the higher the recall of a method is, the lower the recall tends to be. One way of offering more information is to report precision at different levels of recall. Another way is to report average precision, which provides an assessment of how well a system trades precision for recall. The number of all results for computing recall is the number of all relation tuples in the KB.

Relation	RelOnly		Stanf		FIGER		OS		IL	
	F1-top	P-avg	F1-top	P-avg	F1-top	P-avg	F1-top	P-avg	F1-top	P-avg
Musical Artist : album	0.071	0.175	0.079	0.109	0.116	0.203	0.158	0.409	0.115	**0.569**
Musical Artist : record label	0.090	0.182	0.100	0.345	0.179	0.636	0.404	0.758	0.376	**0.926**
Musical Artist : track	0.093	0.109	0.053	0.175	0.104	0.400	0.118	**0.471**	0.114	0.367
Politician : birthplace	0.410	0.594	0.514	0.541	0.496	0.609	0.585	**0.709**	0.516	0.548
Politician : educational institution	0.321	0.387	0.330	0.426	0.366	0.560	0.419	0.719	0.381	**0.831**
Politician : spouse	0.148	0.197	0.152	0.197	0.082	0.309	0.218	**0.319**	0.150	0.181
Business : employees	0.059	0.090	0.097	0.153	0.082	0.325	0.149	0.291	0.133	**0.493**
Business : founders	0.341	0.256	0.462	0.332	0.404	0.542	0.448	0.663	0.429	**0.693**
Education : mascot	0.148	0.362	0.195	0.483	0.226	0.500	0.225	0.506	0.206	**0.585**
Education : city	0.630	0.705	0.711	0.740	0.701	0.770	0.724	0.847	0.690	**0.872**
Film : director	0.383	0.548	0.445	0.603	0.358	0.554	0.439	0.601	0.387	**0.612**
Film : producer	0.149	0.384	0.209	0.395	0.164	0.387	0.198	0.355	0.227	**0.400**
Film : actor	0.246	0.576	0.308	0.633	0.351	0.609	0.342	0.684	0.312	**0.732**
Film : character	0.093	0.123	0.093	0.117	0.180	0.195	0.194	0.298	0.173	**0.319**
Book : author	0.629	0.852	0.703	0.852	0.781	0.878	0.773	0.867	0.781	**0.885**
Book : characters	0.224	0.127	0.193	0.127	0.262	0.328	0.268	0.315	0.231	**0.355**
River : origin	0.175	0.328	0.232	0.493	0.160	0.351	0.256	0.406	0.228	**0.550**
River : mouth	0.336	0.594	0.423	0.564	0.347	0.529	0.488	**0.709**	0.479	0.668

Table 6.5: Results for best model for each relation. Metrics reported are first best precision (P-top), first best recall (R-top), first best F1 (F1-top), all precision (P-all), all recall (P-all), and all average precision (P-avg)(Manning et al., 2008). The number of all results for computing recall is the number of all relation tuples in the KB. The highest P-avg in bold.

Relation	NEC Features	Rel Features
Musical Artist : album	2cont + 1cont + mentfeats + web	parse
Musical Artist : record label	2cont + 1cont + mentfeats + web	parse + 2contword
Musical Artist : track	parse + 2cont + 1cont + mentfeats	parse
Politician : birthplace	2cont + 1cont + mentfeats + web	parse
Politician : educational institution	parse + cont + ment	parse
Politician : spouse	parse + 2cont + 1cont + web	parse
Business : employees	2cont + 1cont + mentfeats + web	parse + 2contword
Business : founders	parse + cont + ment	parse
Education : mascot	parse + 2contwordpos	parse + cont
Education : city	parse + cont + ment	parse + 2contwordpos
Film : director	2cont + 1cont + mentfeats + web	parse + 2contword
Film : producer	parse + cont	parse + 2contwordpos
Film : actor	parse + 2cont + web	parse + 2contwordpos
Film : character	parse + cont + ment	parse + 2contword
Book : author	2cont + 1cont + mentfeats + web	parse
Book : characters	parse + cont + ment	parse
River : origin	2cont + 1cont + mentfeats + web	parse + 2contword
River : mouth	2cont + 1cont + mentfeats + web	parse

Table 6.6: Best feature combination for IL

6.6 Results and Discussion

NEC Features	Rel Features	P-top	R-top	F1-top	P-all	R-all	P-avg
2cont	parse	0.215	0.399	0.28	0.253	0.316	0.381
2cont + 1cont + mentfeats	parse	0.239	0.456	0.313	0.275	0.378	0.441
2cont + 1cont + mentfeats + web	parse	0.248	0.51	0.322	0.276	0.431	0.502
2cont + web	parse	0.204	0.375	0.264	0.244	0.289	0.35
2cont	parse + 2contwordpos	0.236	0.43	0.305	0.275	0.338	0.402
2cont + 1cont + mentfeats	parse + 2contwordpos	0.239	0.456	0.313	0.275	0.378	0.441
2cont + 1cont + mentfeats + web	parse + 2contwordpos	0.248	0.518	0.324	0.275	0.421	0.486
2cont + web	parse + 2contwordpos	0.24	0.402	0.3	0.279	0.305	0.371
2cont	parse + 2contword	0.215	0.394	0.278	0.258	0.309	0.372
2cont + 1cont + mentfeats	parse + 2contword	0.231	0.453	0.295	0.266	0.352	0.43
2cont + 1cont + mentfeats + web	parse + 2contword	0.25	0.54	0.325	0.284	0.433	0.505
2cont + web	parse + 2contword	0.223	0.395	0.285	0.263	0.305	0.373

Table 6.7: Imitation learning results for different NE and relation features, macro average over all relations. Metrics reported are first best precision (P-top), first best recall (R-top), first best F1 (F1-top), all precision (P-all), all recall (P-all), and all average precision (P-avg)(Manning et al., 2008).

6.6.1 Comparison of Models

Overall results in Table 6.4 show that both of the models introduced in this chapter (**IL** and **OS**) outperform the baselines with off-the-shelf supervised NEC (**Stanf**, **FIGER**) for all metrics. Detailed results for different relations (Table 6.5) show that IL outperforms both OS and Base in terms of average precision. FIGER results fall in between Stanf and OS results. For some relations, there is a dramatic improvement by using fine-grained FIGER NE features over coarse-grained Stanford NE features; occasionally FIGER even outperforms OS, as for the relation author. This is because FIGER has a corresponding NE type (see Section 6.4.4).
For most relations, including those whose objects are of type MISC, IL shows a significant improvement in terms of F1 or average precision over OS (Table 6.5). This confirms the hypothesis that separating the NEC and relation extraction stages using imitation learning can achieve a higher precision and recall for non-standard relations than preprocessing sentences with a supervised NEC model. Furthermore, results show that it can also be useful for most standard relations. The main relations for which Stanf, FIGER or OS can have a benefit over IL are those for which entities are easy to classify, specifically LOC NEs, but also PER NEs. This is because, if NEs are easy to classify, a separate NEC is less likely to be useful.

6.6.2 Imitation Learning vs One-Stage

To give more insight into why **IL** is overall more successful than **OS**, common errors made by OS are shown here, along with an explanation of how those errors are prevented by using IL. One example of IL predicting correctly but OS incorrectly is from the following sentence, expressing the director relation:

"In 2010 he appeared in a leading role in <o>Alicia Duffy</o>'s <s>All Good Children</s>."

In that example, the NEC features extracted for <o>Alicia Duffy</o> are not very strong indicators, since neither the object string itself nor the surrounding context give any direct indication for the director relation. The RE features, which are based on the dependency path, are a stronger indicator. Since in the OS model all features are combined, the NEC features overpower the RE features. The IL model, on the other hand, learns a permissive NEC as a first stage, which filters NEs with respect to whether they are generally appropriate for the relation or not, and then leaves the RE to the second stage.

Another example is a sentence for which OS incorrectly predicts the relation author, whereas IL correctly predicts "false":

"<o>Laura</o> and Mary went to school for the first time in Pepin rather than Walnut Grove, which is not included in <s>Little House in the Big Woods</s>."

For this example, OS relation features have small positive weights, which then overall lead to a positive prediction. For IL, the first stage predicts "false", since the one-token string <o>Laura</o> is not a likely candidate for author.

6.6.3 Comparison of Features

All different feature groups have an overall positive effect on the results (see Table 6.7). While low precision, high frequency features improve recall (1cont), they do not always improve precision. Both **OS** and **IL** benefit from high precision, low frequency features, e.g. for author and mouth, the best results are achieved with only sparse parsing features for RE.

Web features improve performance for 10 out of 18 relations. For n-ary relations the is list element feature is very useful because Web pages about musical artists, films or books often contain lists with their attributes, e.g. a Web page about a musical artist typically contains a list with their albums. For relations with persons as objects, is link and is bold is useful because Web pages often highlight persons or provide links to Web pages with more information about them. As an example, for the author relation, the strongest positive Web feature is is in title and the strongest negative feature is is list element. This makes sense since a book is frequently mentioned with its author, which is one of the most important attributes of a book, whereas lists on Web pages about books mention less important attributes, such as the characters.

6.6.4 Overall Comparison

Overall, experiments documented in this chapter showed that using an off-the-shelf NEC as a pre-processing step for distant supervision as done by existing works often causes errors which can be prevented by instead separating NEC and RE with imitation learning. Experiments also showed that using Web features increases precision for NEC. Finally, it is worth noting that the recall for some of the relations is quite low because they only infrequently occur in text, especially in the same sentence as the subject of the relation. These issues can be overcome by performing co-reference resolution (see Section 4), by retrieving more Web pages or improving the information retrieval component of the approach (West et al., 2014) and by combining extractors operating on sentences with other extractors for semi-structured content on Web pages (Carlson et al., 2010a).

6.7 Conclusion and Future Work

This chapter contains experiments addressing one important shortcoming of distant supervision for relation extraction which has been largely ignored by existing work.

The central contribution of this chapter is **a novel method for NERC for distant supervision** using joint learning of NEC and RE with imitation learning. To date, there is very little research on improving NERC for distant supervision to extract relations between non-standard entities such as musical artists and albums. Some research has been done on improving distant supervision by using fine-grained named entity classifiers (Ling and Weld, 2012; Liu et al., 2014) and on using named entity linking for distant supervision (Koch et al., 2014). Liu et al. (2014) train a supervised fine-grained NERC on Wikipedia and show that using those types as entity contraints improves precision and recall for a distantly supervised RE on newswire. However, they assume that labeled training data is available, making it unsuitable for applying distant supervision to domains with relations involving non-standard entity types. Vlachos and Clark (2014b) proposed

a distantly supervised approach for joint learning of NEC and RE with imitation learning for the architecture domain. However, they did not perform experiments on feature selection or named entity candidate identification. Further, they only used two relations in their experiments which involved rather standard entity types and they did not compare against using off-the shelf NEC systems.

The chapter proposes a method for extracting non-standard relations with distant supervision that learns a NEC jointly with relation extraction using imitation learning. The method is inspired by some of the findings in the previous chapter. Those are that unseen NEs, i.e. NEs appearing in the training, but not the testing corpus, make up more than half of the testing instances, and even more for diverse domains such as the Web domain. Unseen NEs are a bottleneck for NERC performance, leading to a substantial drop in performance of 22% on average. This phenomenon occurs even if training and testing corpora belong to the same domain or if large training corpora are used. For the Web corpora used in the study in the previous chapter, performance on them was similar for training data from the same domain or a large training corpus from a different domain. One possibility, which has been researched thoroughly, would be to apply transfer learning methods to try to improve performance, i.e. try to combine a variety of different NERC training corpora and automatically adapt them to the testing domain. But since the base performance for training corpora from the same domain is the same as for using large training corpora from the newswire domain, this possibility has been discarded.

Therefore, a more unusual method is chosen, which has additional benefits: imitation learning only requires labels for outputs (relation labels) but not for intermediate stages (NEC). This means no additional manually labeled training data is neccessary and the method is easy to port to new domains.

The proposed method using the imitation learning algorithm DAgger (Ross et al., 2011) is **thoroughly compared to the state of the art**. The following methods are compared: (1) imitation learning with joint learning of NEC and RE; (2) an internal baseline with a one-stage classification approach using aggregated NEC and RE features; (3) an internal one-stage classification baseline which only uses RE features; (4) the internal baseline aggregated with Stanford NER (Finkel et al., 2005) and FIGER (Ling and Weld, 2012) NE labels as features to achieve a soft NE type constraint. The experiments **show that the proposed method improves over the state of the art**. The proposed imitation learning approach outperforms models with supervised NEC for relations involving non-standard entities as well as relations involving persons, locations and organisations. An increase of 4 points in average precision over a simple one-stage classification model is achieved, and an increase in 10 points and 19 points over baselines with FIGER and Stanford NE labels.

A further contribution of this chapter are thorough experiments regarding **NEC and RE features** and the proposal of **relation candidate identification** strategies. In terms of relation features, low precision, high frequency features such as BoW features are compared against sparse, but high-precision features such as parsing features. Findings are that high frequency features improve recall, but not precision, and that the best results are achieved with only sparse parsing features for relation extraction. For NEC, traditional NEC features such as capitalisation pattern

and context words are tested in addition to Web features. Web features tested are appearances of entities in lists, links to other Web pages, is header or subheader, is bold, is emphasised, is italics, is title and is in title. Those Web features improve performance for 10 out of 18 relations, and improve overall average precision by 7 points.

Particularly useful features for n-ary relations are "is in title" and "is list element", which are used as global features. This is because Web pages about musical artists, films or books often contain lists with their attributes, e.g. a Web page about a musical artist typically contains a list with their albums. Other Web pages are review pages which contain a very detailed review about a particular musical album and therefore contain the name of the album in the title. For relations with persons as objects, such as Book: author, is link and is bold are useful because Web pages often highlight persons or provide links to Web pages with more information about them.

These findings are complementary with existing work which use Web pages for information extraction focusing on using links as local features, particulary in the context of Wikipedia (Wu and Weld, 2010; Presutti et al., 2014). Specialised Web features could also be used to improve other Web search-based distantly supervised relation extraction approaches (Dong et al., 2014; West et al., 2014).

Lastly, a modified version of the **corpus** in Chapter 4 is made available. The corpus contains automatically labeled training and testing examples for 18 different relations and is made available in the same format as Angeli et al. (2014b)'s distant supervision data for easy reuse.

Note that the results presented in this chapter are not directly comparable to the methods presented in Chapter 4, among other reasons, because different evaluation metrics are reported. However, the one-stage classification model reported in this chapter is similar to the baseline model without training data selction reported in Chapter 4. The improvements the approach for joint NEC and RE proposed in this chapter brings are therefore very likely orthogonal to the improvements the training data selection methods proposed in Chapter 4 bring.

In future work, the proposed approach could be combined with other approaches to solve typical issues arising in the context of distant supervision, such as dealing with overlapping relations (Hoffmann et al., 2011), improving heuristic labelling of sentences (Takamatsu et al., 2012) or dealing with incomplete knowledge bases (Min et al., 2013).

Chapter 7

Conclusions

7.1 Conclusions

This thesis has considered how to extract relations from the Web without manually labelled data. To this end, an automatic labelling method called distant supervision was investigated, which assumes the presence of a partly populated knowledge base and utilises the relation tuples already contained in the knowledge base to label training data for relation extraction. A number of research questions with respect to distant supervision have been identified and investigated.

- Setting and Evaluation: how can a distant supervision system for the Web be evaluated?
- Selecting Training Instances: how can incorrectly labelled training data be identified automatically? Can knowledge contained in the knowledge base be exploited for assessing this? How can noisy training data be discarded automatically using inexpensive statistical methods?
- Named Entity Recognition of Diverse NEs: what are the main reasons for low NERC performance in diverse domains, such as the Web and social media? What lessons can be learned from this with respect to recognising NEs for distant supervision?
- NERC for Distant Supervision: how can NERC for RE be performed without the need for additional manual labelled training examples? How can NERC and RE be modelled in a joint way which avoids the problem of error propagation pipeline architectures face?
- Feature Extraction: are features based on markup on Web pages helpful for distantly supervised relation extraction? For a joint modelling approach of NERC and RE, what kinds of NERC and RE features are helpful? Are highly frequent or sparse features better? Shallow or parsing-based ones?
- Selecting Testing Instances and Combining Predictions: are testing instances for relation extraction obtained by preprocessing sentences with co-reference resolution helpful for improving knowledge base population performance? What methods achieve a high performance for combining predictions for knowledge base population?

In the rest of this section, the contributions of this thesis are summarised.

7.1.1 Setting and Evaluation

One of the first important things that need to be considered for every task is: how can it be evaluated (see Section 3.2.1)? Typically, relation extraction is evaluated on manually labelled test data. For the benchmarks ACE 2005 (Walker et al., 2006) and OntoNotes (Hovy et al., 2006), annotations of relations are given, i.e. stand-off annotation files provide character offsets for entity mentions and state what relations hold between those entities. Other evaluation scenarios, such as the TAC KBP evaluations (Surdeanu and Ji, 2014) have the goal of populating knowledge base, and as such measure performance for extraction relations instead of making predictions for individual relation mentions.

In the case of distant supervision for Web relation extraction, evaluation poses a substantial challenge. How can a distantly supervised Web relation extraction approach be evaluated, given that there are no benchmarks available for that task? Two possibilites for evaluation are investigated. The first is a sentence-level manual evaluation, to evaluate extraction quality on relation mention level, as it is the setting for ACE and OntoNotes. However, testing sentences are not annotated beforehand, but rather, extractions are annotated. For this, extracted relation mentions are ranked by confidence in descending order and the correctness of the top 10% of relation mentions is evaluated manually. This type of evaluation is investigated in Section 4. The benefit of this type of evaluation is that it gives a good estimate of precision, however, recall in such an evaluation setting is not very meaningful. For this reason, a second evaluation scenario is considered, for the task of knowledge base completion. For the knowledge base completion task, similar to the TAC KBP evaluations, the input is a knowledge base which is populated with entities and partly populated with relation, and the task is to fill in the missing relations. To fill those in, relations are extracted and predictions of relation mentions are combined. For this setting, a hold-out evaluation can be performed: part of the knowledge base can be used for training and the other part for testing. What is measured is the performance of the relation extraction approach at recreating the knowledge base. Such an evaluation scenario is both more realistic, since it imitates a possible application, and it also does not require manual effort at testing time for labelling data.

Comparing the two evaluation settings, the results for manual sentence-level and automatic knowledge base completion evaluation are very similar. Thus, it can be concluded that an automatic knowledge base completion evaluation is at least equally suited, if not favourable to a manual sentence-level evaluation. Consequently in Section 6, an evaluation is only performed on instance-level.

Considering there is no benchmark available for distantly supervised Web-based relation extraction, a new corpus was created, and a new setting for automatically labelling data with relations proposed in Chapter 4. The setting simulates user queries of the type “What albums were released by The Beatles”, which are then issued against a search engine for retrieving Web pages. Relation triples with subject “The Beatles” from the background knowledge base can then be used to automatically annotate the retrieved Web pages and in turn be used as training data. For testing, a held-out-part of the corpus created in this way is used.

The benefit of this approach is that it is efficient since it does not require many comparisons with the knowledge base. By using search queries constructed from entities and relation names in the knowledge base, the assumption used in this thesis is that the retrieved Web pages indeed contain information about those entities and relations. A further contribution of this to the research community is a new automatically annotated corpus for Web-based distant supervision.

In terms of the aims stated in Section 3.2.1, the research performed meets the basic goals. What could be improved is to thoroughly evaluate the search-based component of the proposed setting to understand how relevant the Web pages retrieved are and mesure if using a search-based approach results in a reduction in error rate for distant supervision compared to considering a large Web crawl. Further, the run time for creating distant supervision corpus could be measured. While the proposed setting requires fewer comparisons with the background knowledge base than previously used ones, it would be interesting to study by how much the run time is reduced, taking into account that using a search engine also requires computational effort.

7.1.2 Selecting Training Instances

Although distant supervision allows one to automatically produce training data without the need for manual labelling, this also has downsides, one of them being that automatically labelled training data can be noisy. Research questions are therefore: how can incorrectly labelled training data be identified automatically? Can knowledge contained in the knowledge base be exploited for assessing this? How can noisy training data be discarded automatically using inexpensive statistical methods?

These research aims are introduced in Section 3.3 and investigated in Chapter 4. The main goal of such training data selection methods is to increase precision for knowledge base population. The approach assesses how likely it is for objects of relations to be ambiguous at training time and then discard them if it is very likely that they are ambiguous. Ambiguity is measured in terms of the number of senses, which is the number of unique resources representing a lexicalisation. The number of unique resources is retrieved based on the knowledge base Freebase.

Two types of ambiguity are measured and evaluated: ambiguity of objects of relations and ambiguity across classes. For the first, a very simple heuristic is used: if a subject is related to two objects with the same lexicalisation, both relations are discarded as training data (*Unam*). For the latter, based on existing entries in the knowledge base, it is estimated how ambiguous a lexicalisation of an object is compared to other lexicalisations of objects of the same relation by viewing the number of lexicalisations as a frequence distribution. If the lexicalisation of an object has more senses than $n\%$ of other objects, it is discarded. This is tested for $n = 25$, $n = 50$ and $n = 75$ (*Stat*). Since this might discard too many lexicalisations for some objects, a refinement of Stat is to only discard lexicalisations of objects that have at least 3 lexicalisations (*StatRes*). In addition, stop words as discarded, as they are generally considered to be ambiguous (*Stop*).

Findings are that those measures indeed manage to remove false positives and thus improve precision for knowledge base population. The highest precision is achieved with *Unam*, *Stop* and *Stat* for $n = 75$. The biggest improvement in precision comes from *Stat*. Discarding stop words does not add much since most of them are captured by *Unam*. However, using*StatRes* instead of

Stat achieves a better trade-off of precision and recall.

While the methods indeed increase precision, this comes at a cost of recall for knowledge base population. What was not studied in the context of those experiments is what role the number of training examples plays with respect to recall for knowledge base population. The assumption of this chapter was that recall could be improved by gathering more Web pages. However, a detailed analysis of the relationship between the number of training examples and recall for knowledge base population when discarding noisy training was not studied. As such, the aims introduced in Section 3.3 were largely achieved, but some of the assumptions made when drawing conclusions could be evaluated more thoroughly.

7.1.3 Named Entity Recognition of Diverse NEs

Relation extraction relies on the task of named entity recognition since named entities often form the arguments of relations. Approaches which study distant supervision for relation extraction typically assume that named entity recognition and classification is solved by using an off-the-shelf NERC such as Stanford NER. What makes NERC for distantly supervised relation extraction so challenging is that, unlike for supervised relation extraction, no named entity annotations are available.

Section 3.4 introduces the research aims of this thesis with respect to named entity recognition and classification. The first one is to analyse and quantify reasons for low NERC performance in diverse genres, such as the Web and social media, which is studied extensively in Chapter 5. NLP tasks are typically more challenging for noisy text which contains spelling mistakes and dialectical and informal usage. Such noisy text can be found on Web pages, in tweets or blogs. For the purpose of quantifying reasons for low NERC performance in diverse genres, popular NERC corpora including MUC 7, ConLL, subcorpora of the ACE and OntoNotes corpora which contain Web data, and social media corpora are studied. NERC performance is measured with CRFSuite, Stanford NER and SENNA.

Firstly, by analysing different corpora, it can be observed that they differ widely in terms of size; in terms of how balanced NE type annotations are in the corpus; in terms of what proportion of the texts are NEs; and how often NEs and tokens are repeated. The most balanced corpus in terms of NE types out of the ones studied is the ConLL corpus, which is the most widely used NERC corpus and the one off-the-shelf NERC systems, such as Stanford NER, are tuned on. Traditionally viewed as noisy, corpora such as Twitter corpora and Web corpora have a low repetition of NEs and tokens, but surprisingly also the ConLL corpus, indicating that it is well balanced in terms of stories. In newswire corpora, a large proportion of the text consists of NEs, which indicates high information density. Web, Twitter and telephone conversation corpora on the other hand have low information density.

Out of the NERC approaches studied, SENNA achieves the highest performance across corpora, and is thus the best at generalising from training to testing data. This can mostly be attributed to the approach using word embeddings being trained with deep convolutional neural nets. The default parameters of SENNA achieve balanced precision and recall, while for Stanford NER and CRFSuite, precision is almost twice as high as recall. As expected, there is a correlation between

NERC performance and training corpus size. However, corpus size is not an absolute predictor of F1. The biggest corpus studied is OntoNotes NW, which is almost twice the size of ConLL in terms of NEs. However, the average F1 for CoNLL is the highest one of all corpora. There is an 11 point difference between the F1 on ConLL and OntoNotes NW with SENNA.

Studying NERC on size normalised corpora, it becomes clear that there is also a big difference in performance for corpora of the same genre. Moreover, with training corpora normalised by size, diverse corpora such as Web corpora and social media corpora achieve a similar F1 as newswire corpora, suggesting that annotating more sentences for diverse genres would also dramatically increase F1.

What is found to be a good predictor of F1 is a memorisation baseline, which picks the most frequent NE label for each token sequence in the test corpus as observed in the training corpus. Inspired by this, the proportion of unseen NEs in the test corpus, i.e. NEs which appear in the testing, but not the training corpus, are studied, as well as the performance on seen and unseen NEs only. What can be learned from this is that corpora with a large proportion of unseen NEs tend to have a lower F1, which is due to F1 being much lower for unseen than for seen NEs (about 17 points lower averaged over all NERC methods and corpora). This finally explains why the performance is highest for the ConLL corpus – it contains the lowest proportion of unseen NEs. It also explains the difference in performance between NERC on other corpora. Out of all the possible indicators for high NER F1 studied, this is found to be the most reliable one.

Also studied is the proportion of unseen features per unseen and seen NE portions of different corpora. However, this is found to not be very helpful. The proportion of seen features is higher for seen NEs, as it should be. However, within the seen and unseen NE splits, there is no clear trend indicating if having more seen features helps.

As mentioned above, hand-annotating more training examples is likely to be a straightforward way of improving NERC performance. However, this is costly, which is why it can be useful to study if using larger corpora for training belonging to the same genre, but taken from a different corpus, might be helpful. In the literature and in this chapter, a binary distinction is made between "in-genre" and "out-of-genre" scenarios. Note that in practice, this such a binary distinction does not exist. Rather, genres can be more or less similar to one another.

Findings of this chapter are that substituting original in-genre training corpora with other training corpora for the same genre created at the same time improves performance, and studying how such corpora can be combined with transfer learning strategies might improve performance even further. However, for most corpora, there is a significant drop in performance for out-of-genre training. What is again found to be reliable is to check the memorisation baseline: if results for the out-of-domain memorisation baseline are higher than for in-genre memorisation, than using the out-of-genre corpus for training is likely to be helpful.

The experiments in Chapter 6 fulfill all aims stated in Section 3.4. It could be argued that, although the chapter thoroughly compares against several baselines, it does not incorporate other findings for improving distant supervision, such as multi-instance multi-label learning (Surdeanu et al., 2012), or also the training data selection methods introduced in Section 4. Although the findings in Chapter 6 are orthogonal to improving how training data is selected, it would be

interesting to see how much improvement could be gained from combining all of those findings.

7.1.4 NERC for Distant Supervision

Going back to NERC for relation extraction for Web data, the goal is to improve performance over using a pre-trained NERC, e.g. Stanford NER trained on the ConLL corpus. One possibility, based on the research in Chapter 5, would be to find corpora with a large memorisation performance when applied to the Web corpus used for relation extraction experiments in this thesis. However, finding suitable training corpora is difficult for diverse genres. In addition, training models for different tasks such as NERC and RE in sequence leads to error propagation, i.e. errors made at earlier stages in the pipeline to be propagated to later stages in the pipeline.

The goal of Chapter 6 is to research methods for NERC for RE which does not require manual labelled training examples for NERC and thus avoid the problem of error propagation. In order to do so, the task of relation extraction is first decomposed into three tasks, which can be seen as different stages: named entity boundary recognition (NER), named entity classification (NEC) and relation extraction (RE). The goal is to achieve a high recall at earlier stages, possibly at the loss of precision and have a high precision at the later stages.

For NER, simple heuristics are used with the aim of capturing most NEs. These are based on capitalisation, part of speech tags and markup on Web pages. In a small manual evaluation measuring recall against Stanford NER, it is confirmed that Stanford NER indeed fails to recognise many true NEs which are recognised with those heuristics. For NEs which should be of type MISC (though the type is disregarded for this small experiment and only handled at a later stage), the recall with Stanford NER is only half of the recall of those simple candidate identification heuristics.

Using those NER heuristics, it is then possible to annotate sentences with NE types using a supervised pre-trained model, such as Stanford NER, or also FIGER. The latter assigns fine-grained NE types and has been used in previous distant supervision work to improve over a distant supervision approach with Stanford NER. The NEC annotations are then used as features and a distantly supervised RE can be trained, as existing work and the approach in Chapter 4 have done. To evaluate how much NEC contributes, a baseline for this is to skip the NEC stage and only train a relation extractor with relation features, but no NEC features. Adding NEC features with Stanford NER (*Stanf*) or FIGER (*FIGER*) improves over the baseline with relation features only (*RelOnly*) by 1.5 and 11 points, respectively, as well as by 3 points in F1 and 2.5 points in F1. This shows that NE types are important for relation extraction, and efforts to improve NEC for relation extraction, such as using a fine-grained NEC (FIGER) instead of a coarse-grained NEC (Stanf) already improves results.

As experiments on the above have already been documented by existing work, Chapter 6 further proposes novel methods which improve over the state of the art. One of those methods is to add NEC features to relation features (conclusions about different features are made in Section 7.1.5), called one-stage model (*OS*). This brings an improvement of 7 further points on average precision and 6 points in F1. However, simply adding NEC features to relation features means the model cannot learn to distinguish between the tasks of NEC and RE. Some NEs might be of the right type for the relation, but still not be the correct object of the relation.

Therefore, in addition to this one-stage model, a joint modelling approach for NEC and RE is proposed, which learns two classifiers (NEC and RE) and dependencies between them. After training, the classifiers are applied in sequence. If the NEC stage concludes that the NE is of the right type for the relation, then RE is applied to make the final prediction, otherwise the RE stage is not reached. However, this typically requires training data. The proposal in Chapter 6 is to train these models with imitation learning. Imitation learning has the attraction of being able to model tasks and dependencies between them, but only require labels for the output (RE). This is done by exploring several actions (positive or negative predictions for NEC) and assessing the effect of these on the final output (positive or negative prediction RE) during training. Results show that such a joint modelling approach with imitation learning (*IL*) further improves average precision by 3.6 points over OS.

7.1.5 Feature Extraction

A further aim, introduced in Section 3.5 is to study what features perform well for distantly supervised relation extraction, specifically if features based on markup on Web pages are helpful. These experiments are documented in Chapter 6.

In that chapter, two kinds of features are studied: relation extraction features and NEC features (for relation extraction). For relation extraction, both shallow and parsing-based features are studied. Shallow features include features such as the sequence of part of speech tags between the subject and the object candidate of the relation. Shallow features are studied included as both 1-grams and 2-grams. Parsing features include, e.g., the dependency path and the lexicalised dependency path between the subject and the object of the relation.

For NEC, traditional features based on existing work are studied, such as the mention, part of speech tags of the mention, digit and capitalisation pattern. In addition to that, local and global features based on HTML markup are extracted: is link, is list element, is header or subheader, is bold, is emphasised, is italics, is title or is contained in title.

What is found for RE is that high precision, low frequency features achieve the highest performance; the best results are achieved with only sparse parsing features for RE. The NEC stage benefits from all features. Web features, however, have the biggest impact on the NEC stage: they improve performance for 10 out of 18 relations. What is particularly useful for n-ary relations such as *album* is the *in list* feature, as e.g. lists of albums can often be found on Web pages. Links and bold text are particularly useful for relations with persons such as *author* as objects because Web pages often highlight persons or provide links to Web pages with more information about them.

The experiments on feature extraction fulfill all aims stated for those experiments described in Section 3.5.

7.1.6 Selecting Testing Instances and Combining Predictions

Lastly, after models are trained, testing instances have to be selected and predictions combined for knowledge base population. Aims stated Section 3.6 are to study if it is beneficial for knowledge base population to not only extract relations between two named entities, but, e.g., between a

pronoun or a category referring to a proper noun and a NE.

To do so, three different methods are studied. The first is to resolve co-references in testing documents with Stanford Coref (*CorefS*). As for NERC, the problem with this is that the co-reference resolution component is trained on newswire data. Therefore, in addition to this, two other methods are used. The first one is a simple heuristic. It assumes that if a paragraph contains at least one sentence with the subject of the relation which is to be extracted, then all following sentences should be used as relation candidates for extraction as well (*NoSub*). Further, a co-reference resolution method based on gender and number gazetteers is used. For paragraphs which contain at least one mention of the subject by name, the approach then identifies all sentences which contain a noun phrase or pronoun which could be co-referent with the subject. Noun phrases are collected by using the Freebase type of the subject (e.g. "Film") and finding synonyms, hypernyms and hypernyms for those using Wikipedia redirection pages and WordNet. Sentences containing such phrases and one other NE (which could be the object) are then used as testing examples (*CorefN*). Further, those noun phrases are looked up in gender and number gazetteers. If a sentence contains one pronoun which agrees with the subject of the relation in number and gender, and one further NE, which could be the object, it is used as a testing instance (*CorefP*).

Comparing those methods for knowledge base population, findings are that the best performing one in terms of F1 is CorefP, followed by CorefS, CorefN and NoSub. Performing training data selection, as also described in Chapter 4, in addition to testing data selection improves precision for knowledge base population over only using original training data with co-reference resolution methods.

These are interesting results which further strengthen the argument that using NLP methods trained on newswire for pre-processing does not always work across genres. The experiments fulfill aims stated in Section 3.6; however, the reason for some of the results is not entirely clear. Particularly, would co-reference resolution still improve results for knowledge base population if more Web pages for testing were available?

The second aim stated in Section 3.6 is to research methods for combining predictions. Four different methods of combining instance-level predictions are evaluated. First, a distinction is made between combining predictions with the same surface form in a straightforward way after extraction (*Aggr*) and combining feature vectors of testing instances for the same $< s, o >$ tuples before training (*Comb*). Results show that the former, which is the method more frequently used in distant supervision evaluations, is the better performing one in terms of both precision and recall.

Next, two methods are tested which make use of background knowledge in the knowledge base for assessing which predictions to return. The first one is to assess how many results to return for each subject and relation pair. To do this, the number of objects for each subject and relation pair are counted and and viewed as a frequency distribution to get a cut-off for how many results to return (*Limit*). Next, it is determined from the knowledge base whether objects which are related to the same subject often have relations with the same object lexicalisations, e.g. the origin of a river, which is also a location contained by the same river. If a prediction can be made with a high confidence for one of those relations (e.g. River: origin), and another prediction with a

lower confidence for one of the other relations (e.g. River: contained by) can be found, then it might make sense to return both of those predictions (*Multilab*). Both of those methods improve precision, but do not lead to a better F1 measure than Aggr.

In terms of aims on combining predictions introduced in Section 3.6, first steps have been made, but much more research could be done. The proposed methods only lead to an improvement in precision, but not recall, which again may be partly due to not having enough testing data for knowledge base population. There are evaluation campaigns for combining predictions, e.g. there is a track at TAC dedicated to slot filler validation[1]. It would be interesting to participate with further research methods for this and compare them against related work.

7.2 Future Work and Outlook

7.2.1 Imitation Learning with Deep Learning

One of the contributions of this thesis is a method for joint learning of relations and entity types with imitation learning. Imitation learning only requires output annotation, i.e. relation labels, but not labels for the intermediate tasks, i.e. NE classification, and learns dependencies between the two tasks. That work follows the general research trend of joint learning of different tasks, with methods and their application to NLP tasks such as integer linear programming (Roth and Yih, 2004, 2007; Galanis et al., 2012), markov logic networks (Domingos et al., 2008; Riedel et al., 2009), and more recently imitation learning (Daumé et al., 2009; Ross et al., 2011; Vlachos and Craven, 2011; Vlachos, 2012; Vlachos and Clark, 2014a).

One of the biggest current trends in natural language processing is deep learning after huge successes of deep learning in the image recognition community. Deep learning methods learn a latent representation of text, which to some degree eliminates the need for traditional feature engineering based on the output of existing features, e.g. features based on part of speech tags. It is so popular a trend that the community even asks if there is any more to natural language processing than deep learning or if deep learning is a one-for-all solution for natural language processing tasks (Manning, 2015). As Manning (2015) argues, NLP approaches are always dependent on the problem. There is no one-for-all machine learning solution for natural language processing tasks, different shapes of problems require different solutions. Moreover, while using deep learning for NLP tasks typically leads to a reduction in error rate, improvements are much smaller than in the vision community, suggesting NLP is far from "solved" and can still benefit from other research based on linguistic intuition.

However, it could be researched how some of the recent deep learning trends could be combined with the findings of this thesis. One of the developments is research on embeddings, which map words or pairs of words into a low-dimensional vector space. Embeddings can be learnt in an unsupervised way and can be seen as a compressed representation of words. Unsupervised pre-training for deep learning has been found to be extremely successful (Erhan et al., 2010).

Word embeddings (Bengio et al., 2003; Mikolov et al., 2013; Collobert et al., 2011; Levy and Goldberg, 2014) as well as relation embeddings (Bordes et al., 2011; Lin et al., 2015) have been

[1] http://www.nist.gov/tac/2015/KBP/SFValidation/index.html

successfully used in the context of information extraction (Collobert et al., 2011; Lin et al., 2015). As future work, it could be researched how Web-based, lexical and dependency features, as used in experiments in Chapter 6, could be replaced or augmented with embeddings in the context of imitation learning.

The other development are the (deep) neural networks of different flavours themselves, which have been shown to improve performance for NLP tasks, especially in combination with embeddings (Collobert et al., 2011). In the context of imitation learning, cost-sensitive classification with PA (Crammer et al., 2006) is used as a base classifier in this thesis. However, more complicated cost-sensitive classification algorithms could be studied, e.g. based on neural networks (Kukar and Kononenko, 1998; Zhou and Liu, 2010). Chang et al. (2015a) shows an example of using a neural network base classifier with imitation learning for dependency parsing. They show that this setting outperforms state of the art dependency parsing approaches, which also use neural nets as base classifiers, but not imitation learning.

7.2.2 Distantly Supervised Relation Extraction for New Genres

One of the genres even more diverse than Web pages studied in Section 5 is social media. As described in Section 2.4.5, the idea of distant supervision for relation extraction can also be applied to automatic labelling of tweets. Marchetti-Bowick and Chambers (2012) identify keywords related to political subtopics and make the assumption that if that keyword occurs in that tweet, it is about that topic. More complicated approaches are to perform aspect-based sentiment analysis with distant supervision (Marchetti-Bowick and Chambers, 2012; Go et al., 2009) for which keywords related to political subtopics are used to identify topics and then paired with sentiment gazetteers to distantly label tweets as positive, negative or neutral. Another variant of distant supervision on Twitter for POS tagging and NER called "not-so-distant supervision" (Plank et al., 2014) (and thus only vaguely related to distant supervision for relation extraction) uses links in tweets to get a bigger context for tweets. In a similar spirit, tags from YouTube videos which are linked in tweets can be used to provide some sort of distant supervision for tweet topic classification.

However, even with that work mentioned above, distantly supervised relation extraction has not been applied to tweets. Future research questions could be: do tweets contain useful relations that cannot be extracted from other sources, such as Web pages? How accurate would a heuristic which labels tweets with relations be?

7.2.3 Joint Learning of Additional Stages

As experiments in Chapter 6 have shown, it is possible to jointly train an NEC and RE for distant supervision to improve average precision for knowledge base population. In that setting, NER heuristics are used as a preprocessing step for imitation learning with the two stages NEC and RE. One possible extension could be to integrate NER into imitation learning, i.e. the imitation learning model would then consist of three stages. For learning that stage, the same automatic annotations as already used for RE and NEC could be re-used. This would lead to a more complicated model, which at the first stage would need to consider every token of every sentence

to assess if that token is part of a NE mention.

Further, Chapter 4 has shown that co-reference resolution can improve precision and recall of distantly supervised relation extraction for knowledge base population. Experiments in that chapter were conducted with both a supervised co-reference resolution model (Lee et al., 2013), and, more crucially, simple heuristics based on number and gender gazetteers (Bergsma and Lin, 2006) and synonyms gathered from Wikipedia disambiguation pages and WordNet (Fellbaum, 1998). Research questions related to this could be: could such co-reference heuristics be used successfully in the same way as distant supervision heuristics, to annotate sentences with co-references for training? Could this be integrated in a joint model with NER, NEC and RE for knowledge base population?

Another related information extraction task that is useful for knowledge base population is entity linking. This has partly been explored for distant supervision by Koch et al. (2014), who use an external Wikipedia-based tool for that, and by Fan et al. (2015), who only use Web pages which are connected to an entity via the property `/common/topic/topic_equivalent_webpage` in Freebase.

While the Web search-based approach for distant supervision used in this thesis already results in largely correct entities, no linking as such is performed, which can be useful if there are several entities with the same name of the same type. Entity linking e.g could be achieved by further analysing the retrieved Web pages with coarse grained IE methods such as topic modelling.

7.2.4 Joint Extraction from Different Web Content

Most existing Web information extraction approaches focus on either text, list or table extraction. There are a few approaches which combine those (Shinzato and Torisawa, 2004; Carlson et al., 2010a,b; Govindaraju et al., 2013; Pennacchiotti and Pantel, 2009; Min et al., 2012; Dong et al., 2014), but they do so by training separate classifiers for the different tasks, then applying them to different corpora and combining the results. One of the popular methods for doing so is called ensemble learning (Dietterich, 2000), which is e.g. used in the context of NELL (Carlson et al., 2010a). By considering text, tables and lists in isolation, direct dependencies between the different types of information in a local context cannot be learnt, e.g. features indicating if the same information is contained in free text, lists and tables on the same Web pages. Therefore research could be done on how those tasks could be learnt jointly, which would also include studying what kinds of Web page-level features could be used to model such a joint Web information extraction methods.

7.2.5 Differences in Extraction Performance between Relations

One characteristic of relation extraction that was observed in this thesis is that there are big differences in performance between different relations. The best performing ones evaluated in this thesis are Musical Artist: record label, Book: author and Education: city, with average precision $>$ 0.85 (see Table 6.5), the worst-performing ones on the other hand are Politician: spouse and Film : character (average precision $<$ 0.35). A difference in performance is also reported by others,

e.g. West et al. (2014), who use a larger Web-based training set. They report the best results (MRR $>$ 0.8) for Person: nationality and Person: education, and the lowest (MRR $<$ 0.35) for Person: children, Person: siblings and Person: parents. What is interesting to see is that the more difficult ones seem to be n-ary relations with persons as objects.

This opens the following research question: how can F1 be improved for such n-ary relations? Dong et al. (2014) show high results for combining extractors from tables, lists and free text, but is that the only successful strategy? Can F1 for extraction of such n-ary relations also be improved for free text alone, to a degree where the precision is high enough to be suitable for knowledge base population?

Further, findings in Chapter 5 are that unseen NEs, i.e. NEs which appear in the test, but not the training set, are much more difficult to recognise and classify than seen NEs. It is likely that this also applies to relations, i.e. relations with seen NEs as subjects and objects should be more likely to be extracted correctly than those with unseen NEs as subjects and objects. This research question and potential solution could also be investigated.

7.3 Final Words

This thesis investigated the research stream of distant supervision for the task of populating knowledge bases with relations extracted from Web pages. It has made several contributions to research by investigating factors that influence the performance of Web-based distant supervision approaches. Most notably, these include researching methods for selecting training and testing data for Web-based distant supervision based on computationally inexpensive methods; researching reasons for NERC failure in diverse genres; and proposing a method for jointly learning a NERC and a RE with imitation which does not rely on manually labelled data.

While the goal of the research was relation extraction from the Web, the novel methods researched could also be applied to other genres and extended to other information extraction tasks. Similarly, the challenges faced when researching a task for a genre for which no suitable gold standard exists hold across genres and tasks.

Automatically processing natural language is a challenging research area. Through my thesis I learned the importance of thoroughly analysing the underlying data, of empirical experiment design and error analysis. Conducting experiments taught me that there is no "one-for-all" solution when it comes to which methods work. Finally, I have come to appreciate fully the importance of being part of a research community and reviewing and scrutinising each others' ideas.

It is hoped that the research of this thesis will inspire others to investigate open research questions in the area of relation extraction and knowledge base population, leading to more precise, more broadly applicable and faster approaches.

Bibliography

Abbeel, P. and A. Y. Ng (2004), "Apprenticeship Learning via Inverse Reinforcement Learning." In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, 1–, ACM, New York, NY, USA. 97

Agichtein, E. and L. Gravano (2000), "Snowball: Extracting relations from large plain-text collections." In *Proceedings of the Fifth ACM Conference on Digital Libraries* (P. Nürnberg, D. Hicks, and R. Furuta, eds.), DL '00, 85–94, ACM, New York, NY, USA. 15

Alfonseca, E., K. Filippova, J.-Y. Delort, and G. Garrido (2012), "Pattern Learning for Relation Extraction with a Hierarchical Topic Model." In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2* (H. Li, C.-Y. Lin, M. Osborne, G. G. Lee, and J. C. Park, eds.), ACL '12, 54–59, Association for Computational Linguistics, Stroudsburg, PA, USA. 23, 25, 38, 39

Angeli, G., S. Gupta, M. J. Premkumar, C. D. Manning, C. Ré, J. Tibshirani, J. Y. Wu, S. Wu, and C. Zhang (2014a), "Stanford's Distantly Supervised Slot Filling Systems for KBP 2014." In *Proceedings of the Seventh Text Analysis Conference (TAC 2014)*, NIST. 22

Angeli, G., J. Tibshirani, J. Wu, and C. D. Manning (2014b), "Combining Distant and Partial Supervision for Relation Extraction." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (A. Moschitti, B. Pang, and W. Daelemans, eds.), 1556–1567, Association for Computational Linguistics, Doha, Qatar. 23, 27, 39, 97, 117

Arnold, A., R. Nallapati, and W. W. Cohen (2008), "Exploiting Feature Hierarchy for Transfer Learning in Named Entity Recognition." In *Proceedings of ACL-08: HLT* (J. D. Moore, S. Teufel, J. Allan, and S. Furui, eds.), 245–253, Association for Computational Linguistics, Columbus, Ohio. 40

Attardi, G. (2015), "DeepNL: a Deep Learning NLP pipeline." In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing* (P. Blunsom, S. Cohen, P. Dhillon, and P. Liang, eds.), 109–115, Association for Computational Linguistics, Denver, Colorado. 78

Augenstein, I. (2014a), "Joint Information Extraction from the Web using Linked Data." In *International Semantic Web Conference (2)* (P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. A. Knoblock, D. Vrandecic, P. T. Groth, N. F. Noy, K. Janowicz, and C. A. Goble, eds.), volume 8797 of *Lecture Notes in Computer Science*, 505–512, Springer, Riva del Garda, Italy. 6

Augenstein, I. (2014b), "Seed Selection for Distantly Supervised Web-Based Relation Extraction." In *Proceedings of the Third Workshop on Semantic Web and Information Extraction* (D. Maynard, M. van Erp, and B. Davis, eds.), 17–24, Association for Computational Linguistics and Dublin City University, Dublin, Ireland. 6, 45, 47

Augenstein, I., L. Derczynski, and K. Bontcheva (2015a), "Generalisation in Named Entity Recognition: A Quantitative Analysis." *Computer Speech & Language*. Under review. 7, 70

Augenstein, I., A. Gentile, B. Norton, Z. Zhang, and F. Ciravegna (2013), "Mapping Keywords to Linked Data Resources for Automatic Query Expansion." In *The Semantic Web: ESWC 2013 Satellite Events* (P. Cimiano, M. Fernández, V. Lopez, S. Schlobach, and J. Völker, eds.), volume 7955 of *Lecture Notes in Computer Science*, 101–112, Springer Berlin Heidelberg. 7

Augenstein, I., D. Maynard, and F. Ciravegna (2014), "Relation Extraction from the Web using Distant Supervision." In *Knowledge Engineering and Knowledge Management* (K. Janowicz, S. Schlobach, P. Lambrix, and E. Hyvönen, eds.), volume 8876 of *Lecture Notes in Computer Science*, 26–41, Springer, Linköping, Sweden. 7, 45, 47

Augenstein, I., D. Maynard, and F. Ciravegna (2016a), "Distantly Supervised Web Relation Extraction for Knowledge Base Population." *Semantic Web Journal*, 7. 7, 45, 47

Augenstein, I., T. Rocktäschel, A. Vlachos, and K. Bontcheva (2016b), "Stance Detection with Bidirectional Conditional Encoding." *CoRR*, abs/1606.05464. 7

Augenstein, I., A. Vlachos, and K. Bontcheva (2016c), "USFD: Any-Target Stance Detection on Twitter with Autoencoders." In *Proceedings of the International Workshop on Semantic Evaluation* (S. M. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, eds.), SemEval '16, San Diego, California. 7

Augenstein, I., A. Vlachos, and D. Maynard (2015b), "Extracting Relations between Non-Standard Entities using Distant Supervision and Imitation Learning." In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 747–757, Association for Computational Linguistics, Lisbon, Portugal. 7, 95

Bach, N. and S. Badaskar (2007), "A Review of Relation Extraction." *Language Technologies Institute, Carnegie Mellon University*. 9, 10

Baldwin, T., Y.-B. Kim, M. C. de Marneffe, A. Ritter, B. Han, and W. Xu (2015), "Shared Tasks of the 2015 Workshop on Noisy User-generated Text: Twitter Lexical Normalization and Named Entity Recognition." In *Proceedings of the Workshop on Noisy User-generated Text* (W. Xu, B. Han, and A. Ritter, eds.), 126–135, Association for Computational Linguistics, Beijing, China. 70, 71, 72

Bellare, K. and A. McCallum (2007), "Learning extractors from unlabeled text using relevant databases." In *The Sixth International Workshop on Information Integration on the Web* (U. Nambiar and Z. Nie, eds.), AAAI Press. 21

Bengio, Y. (2012), "Deep Learning of Representations for Unsupervised and Transfer Learning." *Unsupervised and Transfer Learning Challenges in Machine Learning*, 7, 19. 81

Bengio, Y., R. Ducharme, P. Vincent, and C. Janvin (2003), "A Neural Probabilistic Language Model." *Journal of Machine Learning Research (JMLR)*, 3, 1137–1155. 127

Bergsma, S. and D. Lin (2006), "Bootstrapping Path-Based Pronoun Resolution." In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association of Computational Linguistics* (N. Calzolari, C. Cardie, and P. Isabelle, eds.), The Association for Computer Linguistics, Jeju Island, Korea. 51, 66, 129

Bizer, C., J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann (2009), "DBpedia-A crystallization point for the Web of Data." *Web Semantics: Science, Services and Agents on the World Wide Web*, 7, 154–165. 21

Blomqvist, E., Z. Zhang, A. L. Gentile, I. Augenstein, and F. Ciravegna (2013), "Statistical Knowledge Patterns for Characterising Linked Data." In *Proceedings of 4th Workshop on Ontology and Semantic Web Patterns*. 7

Blum, A. and P. Langley (1997), "Selection of Relevant Features and Examples in Machine Learning." *Artificial Intelligence*, 97, 245–271. 46

Bollacker, K., C. Evans, P. Paritosh, T. Sturge, and J. Taylor (2008), "Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge." In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 1247–1250, ACM, New York, NY, USA. 1, 13, 21, 48, 71, 73

Bordes, A., J. Weston, R. Collobert, and Y. Bengio (2011), "Learning structured embeddings of knowledge bases." In *Proceedings of the Twenty-Fifth Conference on Artificial Intelligence (AAAI)* (W. Burgard and D. Roth, eds.), San Francisco, California, USA. 127

Bouchard, G., T. Trouillon, J. Perez, and A. Gaidon (2015), "Accelerating Stochastic Gradient Descent via Online Learning to Sample." *CoRR*, abs/1506.09016. 46

Brin, S. (1999), "Extracting Patterns and Relations from the World Wide Web." In *The World Wide Web and Databases* (P. Atzeni, A. Mendelzon, and G. Mecca, eds.), 172–183, Springer. 15, 16

Brown, P. F., P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai (1992), "Class-based n-gram models of natural language." *Computational linguistics*, 18, 467–479. 78

Bunescu, R. and M. Pasça (2006), "Using Encyclopedic Knowledge for Named Entity Disambiguation." In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, volume 6, 9–16, Trento, Italy. 1, 42

Carlson, A., J. Betteridge, B. Kisiel, B. Settles, E. R. H. Jr., and T. M. Mitchell (2010a), "Toward an Architecture for Never-Ending Language Learning." In *Proceedings of the Twenty-Fourth*

AAAI Conference on Artificial Intelligence (M. Fox and D. Poole, eds.), AAAI Press, Palo Alto, California, USA. 16, 17, 22, 28, 115, 129

Carlson, A., J. Betteridge, R. C. Wang, E. R. Hruschka, Jr., and T. M. Mitchell (2010b), "Coupled Semi-supervised Learning for Information Extraction." In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, 101–110, ACM, New York, NY, USA. 17, 43, 129

Chang, K.-W., H. He, H. I. Daumé, and J. Langford (2015a), "Learning to Search for Dependencies." *arXiv preprint arXiv:1503.05615*. 99, 128

Chang, K.-W., A. Krishnamurthy, A. Agarwal, H. I. Daumé, and J. Langford (2015b), "Learning to search better than your teacher." In *Proceedings of The 32nd International Conference on Machine Learning (ICML)*, 2058—2066, Journal of Machine Learning Research (JMLR). 97

Chang, M.-W., L.-A. Ratinov, N. Rizzolo, and D. Roth (2008), "Learning and inference with constraints." In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI)* (D. Fox and C. P. Gomes, eds.), 1513–1518, AAAI Press, Chicago, Illinois, USA. 24

Cherry, C. and H. Guo (2015), "The Unreasonable Effectiveness of Word Representations for Twitter Named Entity Recognition." In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (R. Mihalcea, J. Chai, and A. Sarkar, eds.), 735–745, Association for Computational Linguistics, Denver, Colorado. 71, 72

Chinchor, N. A. (1998), "Proceedings of the Seventh Message Understanding Conference (MUC-7) named entity task definition." In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Fairfax, VA. 71, 72

Chiticariu, L., R. Krishnamurthy, Y. Li, F. Reiss, and S. Vaithyanathan (2010), "Domain Adaptation of Rule-Based Annotators for Named-Entity Recognition Tasks." In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (H. Li and L. M'arquez, eds.), 1002–1012, Association for Computational Linguistics, Cambridge, MA. 71

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa (2011), "Natural Language Processing (Almost) from Scratch." *Journal of Machine Learning Research (JMLR)*, 999888, 2493–2537. 40, 78, 127, 128

Crammer, K., O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer (2006), "Online Passive-Aggressive Algorithms." *Journal of Machine Learning Research (JMLR)*, 7, 551–585. 104, 128

Craven, M., J. Kumlien, et al. (1999), "Constructing Biological Knowledge Bases by Extracting Information from Text Sources." In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology* (T. Lengauer, R. Schneider, P. Bork, D. Brutlag, J. Glasgow, H.-W. Mewes, and R. Zimmer, eds.), volume 1999, 77–86, AAAI Press, Palo Alto, California, USA. 2, 18, 21

Cunningham, H., D. Maynard, K. Bontcheva, and V. Tablan (2002), "GATE: an Architecture for Development of Robust HLT applications." In *Proceedings of ACL*, 168–175, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA. 71

Dalvi, B. B., W. W. Cohen, and J. Callan (2012), "WebSets: Extracting Sets of Entities from the Web Using Unsupervised Information Extraction." In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, 243–252, ACM, New York, NY, USA. 42

Daumé, H. I. (2007), "Frustratingly Easy Domain Adaptation." In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (A. Zaenen and A. van den Bosch, eds.), 256–263, Association for Computational Linguistics, Prague, Czech Republic. 40, 71, 86

Daumé, H. I., J. Langford, and D. Marcu (2009), "Search-based Structured Prediction." *Machine Learning*, 75, 297–325. 97, 99, 127

Demetriou, G., R. J. Gaizauskas, H. Sun, and A. Roberts (2008), "ANNALIST - ANNotation ALIgnment and Scoring Tool." In *LREC*, European Language Resources Association. 64

Derczynski, L., I. Augenstein, and K. Bontcheva (2015a), "USFD: Twitter NER with Drift Compensation and Linked Data." In *Proceedings of the Workshop on Noisy User-generated Text* (W. Xu, B. Han, and A. Ritter, eds.), 48–53, Association for Computational Linguistics, Beijing, China. 7, 96

Derczynski, L. and S. Chester (2016), "Generalised Brown clustering and roll-up feature generation." In *Proceedings of the annual conference of the Association for Advancement of Artificial Intelligence*, AAAI. 78

Derczynski, L., D. Maynard, N. Aswani, and K. Bontcheva (2013), "Microblog-genre Noise and Impact on Semantic Annotation Accuracy." In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT '13, 21–30, ACM, New York, NY, USA. 40, 70, 72

Derczynski, L., D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, and K. Bontcheva (2015b), "Analysis of Named Entity Recognition and Linking for Tweets." *Information Processing and Management*, 51, 32–49. 70, 71, 72

Dietterich, T. G. (2000), "Ensemble Methods in Machine Learning." In *Multiple Classifier Systems*, 1–15, Springer Berlin Heidelberg. 129

Doddington, G. R., A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel (2004), "The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation." In *LREC*, European Language Resources Association. 2, 47

Domingos, P., S. Kok, D. Lowd, H. Poon, M. Richardson, and P. Singla (2008), "Markov logic." In *Probabilistic Inductive Logic Programming*, 92–117. 41, 96, 127

Dong, X., E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang (2014), "Knowledge Vault: A Web-scale approach to probabilistic knowledge fusion." In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 601–610, ACM, New York, NY, USA. 1, 2, 3, 22, 117, 129, 130

Eisenstein, J. (2013), "What to do about bad language on the internet." In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (L. Vanderwende, H. I. Daumé, and K. Kirchhoff, eds.), 359–369, Association for Computational Linguistics, Atlanta, Georgia. 70

Erhan, D., Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio (2010), "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research (JMLR)*, 11, 625–660. 127

Etzioni, O., M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates (2004), "Web-scale Information Extraction in KnowItAll." In *Proceedings of the 13th International Conference on World Wide Web* (S. Feldman, M. Uretsky, M. Najork, and C. Wills, eds.), ACM, Rio de Janeiro, Brazil. 2, 16, 22

Exner, P., M. Klang, and P. Nugues (2015), "A Distant Supervision Approach to Semantic Role Labeling." In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics* (M. Palmer, G. Boleda, and P. Rosso, eds.), 239–248, Association for Computational Linguistics, Denver, Colorado. 29

Fader, A., S. Soderland, and O. Etzioni (2011), "Identifying Relations for Open Information Extraction." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, and S. Bethard, eds.), 1535–1545, Association for Computational Linguistics, Seattle, Washington, USA. 17, 18

Fader, A., L. Zettlemoyer, and O. Etzioni (2014), "Open Question Answering Over Curated and Extracted Knowledge Bases." In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 1156–1165, ACM, New York, NY, USA. 1

Fan, M., Q. Zhou, and T. F. Zheng (2015), "Distant Supervision for Entity Linking." *CoRR*, abs/1505.03823. 31, 129

Fellbaum, C., ed. (1998), *Wordnet, an Electronic Lexical Database*. Language, Speech, and Communication, MIT Press, Cambridge, Massachusetts, USA. 14, 51, 66, 129

Finin, T., W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze (2010), "Annotating Named Entities in Twitter Data with Crowdsourcing." In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* (C. Callison-Burch and M. Dredze, eds.), 80–88, Association for Computational Linguistics, Los Angeles. 70, 71, 73

Finkel, J. R., T. Grenager, and C. Manning (2005), "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling." In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)* (K. Knight, H. T. Ng, and K. Oflazer, eds.), 363–370, Association for Computational Linguistics, Ann Arbor, Michigan. 4, 28, 40, 58, 78, 116

Forman, G. and I. Cohen (2004), "Learning from Little: Comparison of Classifiers Given Little Training." In *PKDD '04: Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 161–172, Springer-Verlag New York, Inc., New York, NY, USA. 80

Fromreide, H., D. Hovy, and A. Søgaard (2014), "Crowdsourcing and annotating NER for Twitter #drift." In *Proceedings of LREC* (N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, eds.), 2544–2547, European Language Resources Association. 70, 72

Gabbard, R., M. Freedman, and R. Weischedel (2011), "Coreference for Learning to Extract Relations: Yes Virginia, Coreference Matters." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (M. Yuji and M. Rada, eds.), 288–293, Association for Computational Linguistics, Portland, Oregon, USA. 3

Gaizauskas, R. and Y. Wilks (1998), "Information Extraction: Beyond Document Retrieval." *Journal of Documentation*, 54, 70–105. 9

Galanis, D., G. Lampouras, and I. Androutsopoulos (2012), "Extractive Multi-Document Summarization with Integer Linear Programming and Support Vector Regression." In *Proceedings of COLING 2012* (M. Kay and C. Boitet, eds.), 911–926, The COLING 2012 Organizing Committee, Mumbai, India. 41, 127

Geman, S. and D. Geman (1984), "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6, 721–741. 25

Gentile, A. L., Z. Zhang, I. Augenstein, and F. Ciravegna (2013), "Unsupervised Wrapper Induction using Linked Data." In *Proceedings of the 7th International Conference on Knowledge Capture* (V. R. Benjamins, M. d'Aquin, and A. Gordon, eds.), 41–48, ACM, New York, NY, USA. 7, 60

Glorot, X., A. Bordes, and Y. Bengio (2011), "Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach." In *Proceedings of the 28th International Conference on Machine Learning (ICML)* (L. Getoor and T. Scheffer, eds.), 513–520, Omnipress, Bellevue, Washington, USA. 81

Go, A., R. Bhayani, and L. Huang (2009), "Twitter Sentiment Classification using Distant Supervision." *Processing*, 1–6. 128

Gorrell, G., J. Petrak, and K. Bontcheva (2015), "Using @Twitter Conventions to Improve #LOD-Based Named Entity Disambiguation." In *ESWC* (F. Gandon, M. Sabou, H. Sack, C. d'Amato, P. Cudré-Mauroux, and A. Zimmermann, eds.), volume 9088 of *Lecture Notes in Computer Science*, 171–186, Springer. 31

Govindaraju, V., C. Zhang, and C. Ré (2013), "Understanding Tables in Context Using Standard NLP Toolkits." In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (P. Fung and M. Poesio, eds.), 658–664, Association for Computational Linguistics, Sofia, Bulgaria. 129

Grishman, R. and B. Sundheim (1995), "Message Understanding Conference-6: A Brief History." In *Proceedings of COLING*, Association for Computational Linguistics. 2, 4, 11, 69, 72

Grissom, A. I., J. Boyd-Graber, H. He, J. Morgan, and H. I. Daumé (2014), "Don't Until the Final Verb Wait: Reinforcement Learning for Simultaneous Machine Translation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (A. Moschitti, B. Pang, and W. Daelemans, eds.), 1342–1352, Association for Computational Linguistics, Doha, Qatar. 99

Guo, H., H. Zhu, Z. Guo, X. Zhang, X. Wu, and Z. Su (2009), "Domain Adaptation with Latent Semantic Association for Named Entity Recognition." In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (M. Ostendorf, M. Collins, S. Narayanan, D. W. Oard, and L. Vanderwende, eds.), 281–289, Association for Computational Linguistics, Boulder, Colorado. 40

Guo, X., S. P. Singh, H. Lee, R. L. Lewis, and X. Wang (2014), "Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning." In *NIPS* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), 3338–3346. 98

Han, B. and T. Baldwin (2011), "Lexical Normalisation of Short Text Messages: Makn Sens a #twitter." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Y. Matsumoto and R. Mihalcea, eds.), 368–378, Association for Computational Linguistics, Portland, Oregon, USA. 72

Han, X., L. Sun, and J. Zhao (2011), "Collective Entity Linking in Web Text: A Graph-based Method." In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, 765–774, ACM, New York, NY, USA. 42

He, H., H. I. Daumé, and J. Eisner (2013), "Dynamic Feature Selection for Dependency Parsing." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, and S. Bethard, eds.), 1455–1464, Association for Computational Linguistics, Seattle, Washington, USA. 99

Hearst, M. A. (1992), "Automatic Acquisition of Hyponyms from Large Text Corpora." In *In Proceedings of the 14th International Conference on Computational Linguistics*, 539–545. 16

Hodges, P. E., W. E. Payne, and J. I. Garrels (1998), "The Yeast Protein Database (YPD): a curated proteome database for Saccharomyces cerevisiae." *Nucleic Acids Research*, 26, 68–72. 21

Hoffart, J., F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum (2011), "YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages." In *Proceedings of the 20th International Conference Companion on World Wide Web* (S. Sadagopan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, eds.), WWW '11, 229–232, ACM, New York, NY, USA. 22

Hoffmann, R., C. Zhang, X. Ling, L. S. Zettlemoyer, and D. S. Weld (2011), "Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Y. Matsumoto and R. Mihalcea, eds.), 541–550, The Association for Computer Linguistics, Portland, Oregon, USA. 21, 22, 23, 24, 26, 28, 29, 38, 42, 43, 108, 117

Hovy, D., A. Johannsen, and A. Søgaard (2015), "User Review Sites as a Resource for Large-Scale Sociolinguistic Studies." In *Proceedings of the 24th International Conference on World Wide Web* (A. Gangemi, S. Leonardi, and A. Panconesi, eds.), 452–461, International World Wide Web Conferences Steering Committee, ACM. 71

Hovy, E., M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel (2006), "OntoNotes: The 90% Solution." In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers* (R. C. Moore, J. Bilmes, J. Chu-Carroll, and M. Sanderson, eds.), 57–60, Association for Computational Linguistics, New York City, USA. 38, 71, 73, 120

Ji, H. and R. Grishman (2005), "Improving name tagging by reference resolution and relation detection." In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)* (K. Knight, H. T. Ng, and K. Oflazer, eds.), 411–418, Association for Computational Linguistics, Ann Arbor, Michigan. 96

Ji, H. and R. Grishman (2011), "Knowledge Base Population: Successful Approaches and Challenges." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Y. Matsumoto and R. Mihalcea, eds.), 1148–1158, Association for Computational Linguistics, Portland, Oregon, USA. 12

Kate, R. J. and R. Mooney (2010), "Joint Entity and Relation Extraction Using Card-Pyramid Parsing." In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning* (M. Lapata and A. Sarkar, eds.), 203–212, Association for Computational Linguistics, Uppsala, Sweden. 96

Kim, J.-D., T. Ohta, S. Pyysalo, Y. Kano, and J. Tsujii (2009), "Overview of BioNLP'09 Shared Task on Event Extraction." In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task* (T. Jun'ichi, ed.), 1–9, Association for Computational Linguistics, Boulder, Colorado. 30

Koch, M., J. Gilmer, S. Soderland, and D. S. Weld (2014), "Type-Aware Distantly Supervised Relation Extraction with Linked Arguments." In *Proceedings of EMNLP* (A. Moschitti, B. Pang, and W. Daelemans, eds.), 1891–1901, Association for Computational Linguistics, Doha, Qatar. 47, 115, 129

Kozareva, Z. (2006), "Bootstrapping Named Entity Recognition with Automatically Generated Gazetteer Lists." In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop* (J. L. R. Sebastian Padó and V. Seretan, eds.), 15–21, Association for Computational Linguistics, Trento, Italy. 71

Kukar, M. and I. Kononenko (1998), "Cost-Sensitive Learning with Neural Networks." In *13th European Conference on Artificial Intelligence (ECAI)* (H. Prade, ed.), 445–449, Brighton, UK. 128

Lafferty, J. D., A. McCallum, and F. C. N. Pereira (2001), "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data." In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, 282–289, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 78

Lee, D. Y. W. (2001), "Genres, Registers, Text Types, Domains, and Styles Clarifying the Concepts and Navigating a Path the the BNC Jungle." *Language Learning and Technology*, 5, 37–72. 70

Lee, H., A. X. Chang, Y. Peirsman, N. Chambers, M. Surdeanu, and D. Jurafsky (2013), "Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules." *Computational Linguistics*, 39, 885–916. 56, 66, 129

Lendvai, P., I. Augenstein, K. Bontcheva, and T. Declerck (2016a), "Monolingual Social Media Datasets for Detecting Contradiction and Entailment." *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*. 7

Lendvai, P., I. Augenstein, D. Rout, K. Bontcheva, and T. Declerck (2016b), "Algorithms for Detecting Disputed Information: Final Version." 7

Levy, O. and Y. Goldberg (2014), "Neural Eord Embedding as Implicit Matrix Factorization." In *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, eds.), 2177–2185, Curran Associates, Inc. 127

Lewis, D. D., Y. Yang, T. G. Rose, and F. Li (2004), "RCV1: A New Benchmark Collection for Text Categorization Research." *Journal of Machine Learning Research (JMLR)*, 5, 361–397. 50, 78

Li, Q. and H. Ji (2014), "Incremental Joint Extraction of Entity Mentions and Relations." In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 402–412, Association for Computational Linguistics, Baltimore, Maryland. 96

Li, X.-L., B. Liu, and S.-K. Ng (2010), "Negative Training Data Can be Harmful to Text Classification." In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language*

Processing (H. Li and L. M'arquez, eds.), 218–228, Association for Computational Linguistics, Cambridge, MA. 48

Lin, Y., Z. Liu, M. Sun, Y. Liu, and X. Zhu (2015), "Learning Entity and Relation Embeddings for Knowledge Graph Completion." In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence (AAAI)* (B. Bonet and S. Koenig, eds.), 2181–2187, Austin, Texas, USA. 127, 128

Ling, X. and D. S. Weld (2012), "Fine-Grained Entity Recognition." In *Proceedings of the Twenty-Sixth Conference on Artificial Intelligence (AAAI)* (J. Hoffmann and B. Selman, eds.), AAAI Press, Toronto, Ontario, Canada. 22, 23, 28, 29, 32, 38, 41, 102, 115, 116

Liu, X., S. Zhang, F. Wei, and M. Zhou (2011), "Recognizing Named Entities in Tweets." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Y. Matsumoto and R. Mihalcea, eds.), 359–367, Association for Computational Linguistics, Portland, Oregon, USA. 71

Liu, Y., K. Liu, L. Xu, and J. Zhao (2014), "Exploring Fine-grained Entity Type Constraints for Distantly Supervised Relation Extraction." In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (J. Tsujii and J. Hajic, eds.), 2107–2116, Dublin City University and Association for Computational Linguistics, Dublin, Ireland. 23, 28, 32, 38, 41, 109, 115

Locke, B. and J. Martin (2009), "Named entity recognition: Adapting to microblogging." *University of Colorado*. 87

Magdy, W., H. Sajjad, T. El-Ganainy, and F. Sebastiani (2015), "Distant Supervision for Tweet Classification Using YouTube Labels." In *Proceedings of the Ninth International Conference on Web and Social Media (ICWSM)* (M. Cha, C. Mascolo, and C. Sandvig, eds.), 638–641, AAAI Press, Oxford, UK. 30

Manning, C. D. (2015), "Computational Linguistics and Deep Learning." *Computational Linguistics*, 41. 127

Manning, C. D., P. Raghavan, and H. Schtze (2008), *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA. vi, 111, 112, 113

Marchetti-Bowick, M. and N. Chambers (2012), "Learning for Microblogs with Distant Supervision: Political Forecasting with Twitter." In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics* (W. Daelemans, ed.), 603–612, Association for Computational Linguistics, Avignon, France. 29, 30, 128

Mausam, M. Schmitz, S. Soderland, R. Bart, and O. Etzioni (2012), "Open Language Learning for Information Extraction." In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (J. Tsujii, J. Henderson, and M. Pasça, eds.), 523–534, Association for Computational Linguistics, Jeju Island, Korea. 17

Maynard, D., K. Bontcheva, and H. Cunningham (2003), "Towards a Semantic Extraction of Named Entities." In *Recent Advances in Natural Language Processing*, Bulgaria. 71

Maynard, D., A. Funk, and W. Peters (2009), "SPRAT: A Tool for Automatic Semantic Pattern-Based Ontology Population." In *International Conference for Digital Libraries and the Semantic Web*. 71

Mendes, P. N., M. Jakob, A. García-Silva, and C. Bizer (2011), "DBpedia Spotlight: Shedding Light on the Web of Documents." In *Proceedings the 7th International Conference on Semantic Systems (I-SEMANTICS)* (C. Ghidini, A. N. Ngomo, S. N. Lindstaedt, and T. Pellegrini, eds.), I-Semantics '11, 1–8, ACM, Graz, Austria. 1

Mikheev, A., M. Moens, and C. Grover (1999), "Named Entity Recognition without Gazetteers." In *Proceedings of EACL*, 1–8, Association for Computational Linguistics, Bergen, Norway. 71

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013), "Efficient Estimation of Word Representations in Vector Space." *CoRR*. 127

Min, B., R. Grishman, L. Wan, C. Wang, and D. Gondek (2013), "Distant Supervision for Relation Extraction with an Incomplete Knowledge Base." In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (L. Vanderwende, H. D. III, and K. Kirchhoff, eds.), 777–782, The Association for Computational Linguistics, Atlanta, Georgia. 24, 25, 26, 32, 56, 117

Min, B., S. Shi, R. Grishman, and C.-Y. Lin (2012), "Ensemble Semantics for Large-scale Unsupervised Relation Extraction." In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (J. Tsujii, J. Henderson, and M. Pasça, eds.), 1027–1037, Association for Computational Linguistics, Jeju Island, Korea. 129

Mintz, M., S. Bills, R. Snow, and D. Jurafsky (2009), "Distant supervision for relation extraction without labeled data." In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (K.-Y. Su, J. Su, J. Wiebe, and H. Li, eds.), 1003–1011, Association for Computational Linguistics, Suntec, Singapore. vii, 2, 12, 18, 19, 21, 23, 24, 25, 27, 38, 42, 43, 48, 52, 56, 57, 58, 62, 101, 108

Mooney, R. J. and R. C. Bunescu (2005), "Subsequence Kernels for Relation Extraction." In *Proceedings of the 19th Conference on Neural Information Processing Systems (NIPS)*, 171–178, Vancouver, BC. 71

Nadeau, D. and S. Sekine (2007), "A Survey of Named Entity Recognition and Classification." *Journal of Linguisticae Investigationes*, 30, 1–20. 71, 108

Nguyen, T. V. T. and A. Moschitti (2011a), "End-to-End Relation Extraction Using Distant Supervision from External Semantic Repositories." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Y. Matsumoto

and R. Mihalcea, eds.), 277–282, Association for Computational Linguistics, Portland, Oregon, USA. 21

Nguyen, T.-V. T. and A. Moschitti (2011b), "Joint Distant and Direct Supervision for Relation Extraction." In *Proceedings of 5th International Joint Conference on Natural Language Processing* (H. Wang and D. Yarowsky, eds.), 732–740, Asian Federation of Natural Language Processing, Chiang Mai, Thailand. 23, 27, 28, 39

Okazaki, N. (2007), "CRFsuite: a fast implementation of conditional random fields (CRFs)." 78

Palmer, D. D. and D. S. Day (1997), "A Statistical Profile of the Named Entity Task." In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 190–193, Washington, DC, USA. 71, 72, 74

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu (2002), "Bleu: a Method for Automatic Evaluation of Machine Translation." In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, 311–318, Association for Computational Linguistics, Philadelphia, Pennsylvania, USA. 99

Parikh, A. P., H. Poon, and K. Toutanova (2015), "Grounded Semantic Parsing for Complex Knowledge Extraction." In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (R. Mihalcea, J. Chai, and A. Sarkar, eds.), 756–766, Association for Computational Linguistics, Denver, Colorado. 30

Pedro, S. D. and E. R. Hruschka Jr (2012), "Conversing Learning: Active Learning and Active Social Interaction for Human Supervision in Never-Ending Learning Systems." In *Advances in Artificial Intelligence - 13th Ibero-American Conference on AI (IBERAMIA)* (R. F.-F. Juan Pavón, Néstor D. Duque-Méndez, ed.), 231–240, Springer, Cartagena de Indias, Colombia. 17

Pennacchiotti, M. and P. Pantel (2009), "Entity Extraction via Ensemble Semantics." In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (P. Koehn and R. Mihalcea, eds.), 238–247, Association for Computational Linguistics, Singapore. 129

Pershina, M., B. Min, W. Xu, and R. Grishman (2014), "Infusion of Labeled Data into Distant Supervision for Relation Extraction." In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (K. Toutanova and H. Wu, eds.), 732–738, Association for Computational Linguistics, Baltimore, Maryland. 23, 27, 28, 39

Plank, B. and D. Hovy (2015), "Personality traits on Twitter–or–How to get 1,500 personality tests in a week." In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 92–98. 71

Plank, B., D. Hovy, R. McDonald, and A. Søgaard (2014), "Adapting taggers to Twitter with not-so-distant supervision." In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (J. Tsujii and J. Hajic, eds.), 1783–1792, Dublin City University and Association for Computational Linguistics, Dublin, Ireland. 31, 71, 128

Preoţiuc-Pietro, D., S. Volkova, V. Lampos, Y. Bachrach, and N. Aletras (2015), "Studying user income through language, behaviour and affect in social media." *PloS one*, 10, e0138717. 71

Presutti, V., S. Consoli, A. G. Nuzzolese, D. R. Recupero, A. Gangemi, I. Bannour, and H. Zargayouna (2014), "Uncovering the Semantics of Wikipedia Pagelinks." In *EKAW* (K. Janowicz, S. Schlobach, P. Lambrix, and E. Hyvönen, eds.), volume 8876 of *Lecture Notes in Computer Science*, 413–428, Springer, Linköping, Sweden. 117

Ratinov, L. and D. Roth (2009), "Design Challenges and Misconceptions in Named Entity Recognition." In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)* (S. Stevenson and X. Carreras, eds.), 147–155, Association for Computational Linguistics, Boulder, Colorado. 71

Riedel, S., H.-W. Chun, T. Takagi, and J. Tsujii (2009), "A Markov Logic Approach to Bio-Molecular Event Extraction." In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task* (J. Tsujii, ed.), 41–49, Association for Computational Linguistics, Boulder, Colorado. 41, 127

Riedel, S., L. Yao, and A. McCallum (2010), "Modeling Relations and Their Mentions without Labeled Text." In *Proceedings of the 2010 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (3)* (J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, eds.), volume 6323 of *Lecture Notes in Computer Science*, 148–163, Springer, Barcelona, Catalonia, Spain. 21, 22, 23, 24, 25, 26, 27, 29, 38, 39

Riedel, S., L. Yao, A. McCallum, and B. M. Marlin (2013), "Relation Extraction with Matrix Factorization and Universal Schemas." In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (L. Vanderwende, H. I. Daumé, and K. Kirchhoff, eds.), 74–84, Association for Computational Linguistics, Atlanta, Georgia. 26, 32

Rieser, V. and O. Lemon (2010), "Natural Language Generation as Planning under Uncertainty for Spoken Dialogue Systems." In *Empirical Methods in Natural Language Generation* (E. Krahmer and M. Theune, eds.), volume 5790 of *Lecture Notes in Computer Science*, 105–120, Springer. 98

Ritter, A., S. Clark, Mausam, and O. Etzioni (2011), "Named Entity Recognition in Tweets: An Experimental Study." In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (R. Barzilay and M. Johnson, eds.), 1524–1534, Association for Computational Linguistics, Edinburgh, Scotland, UK. 69, 70, 71, 73

Ritter, A., L. Zettlemoyer, O. Etzioni, et al. (2013), "Modeling Missing Data in Distant Supervision for Information Extraction." *Transactions of the Association for Computational Linguistics*, 1, 367–378. 26

Roller, R. and M. Stevenson (2014), "Self-Supervised Relation Extraction using UMLS." In *Proceedings of the 5th International Conference of the CLEF Initiative* (E. Kanoulas, M. Lupu,

P. D. Clough, M. Sanderson, M. M. Hall, A. Hanbury, and E. G. Toms, eds.), volume 8685 of *Lecture Notes in Computer Science*, 116–127, Springer, Sheffield, UK. 21, 23, 31, 38

Ross, S., G. J. Gordon, and D. Bagnell (2011), "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning." In *AISTATS* (G. J. Gordon, D. B. Dunson, and M. Dudík, eds.), volume 15 of *JMLR Proceedings*, 627–635, Journal of Machine Learning Research (JMLR). 41, 42, 93, 96, 99, 100, 102, 103, 104, 116, 127

Roth, B., T. Barth, G. Chrupała, M. Gropp, and D. Klakow (2014), "RelationFactory: A Fast, Modular and Effective System for Knowledge Base Population." In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics* (S. Wintner, M. Tadić, and B. Babych, eds.), 89–92, Association for Computational Linguistics, Gothenburg, Sweden. 22

Roth, B., T. Barth, M. Wiegand, and D. Klakow (2013), "A Survey of Noise Reduction Methods for Distant Supervision." In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction* (F. Suchanek, S. Riedel, S. Singh, and P. P. Talukdar, eds.), 73–78, ACM, New York, NY, USA. 19, 24

Roth, B., G. Chrupala, M. Wiegand, M. Singh, and D. Klakow (2012), "Generalizing from Freebase and Patterns using Cluster-Based Distant Supervision for TAC KBP Slotfilling 2012." In *Proceedings of the Fifth Text Analysis Conference (TAC)*, NIST, Gaithersburg, Maryland, USA. 22

Roth, B. and D. Klakow (2013a), "Combining Generative and Discriminative Model Scores for Distant Supervision." In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, and S. Bethard, eds.), 24–29, Association for Computational Linguistics, Seattle, Washington, USA. 39

Roth, B. and D. Klakow (2013b), "Feature-Based Models for Improving the Quality of Noisy Training Data for Relation Extraction." In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management* (Q. He, A. Iyengar, W. Neijdl, J. Pei, and R. Rastogi, eds.), CIKM '13, 1181–1184, ACM, New York, NY, USA. 24

Roth, D. and W.-T. Yih (2004), "A Linear Programming Formulation for Global Inference in Natural Language Tasks." In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)* (H. T. Ng and E. Riloff, eds.), 1–8, Association for Computational Linguistics, Boston, Massachusetts, USA. 41, 96, 127

Roth, D. and W.-T. Yih (2007), "Global Inference for Entity and Relation Identification via a Linear Programming Formulation." *Introduction to Statistical Relational Learning*, 553–580. 41, 96, 127

Rowe, M., M. Stankovic, A. Dadzie, B. Nunes, and A. Cano (2013), "Making Sense of Microposts (#MSM2013): Big things come in small packages." In *Proceedings of the International Conference on World Wide Web - Workshops*. 70, 71, 73

Rowe, M., M. Stankovic, and A.-S. Dadzie (2015), "#Microposts2015 – 5th Workshop on 'Making Sense of Microposts': Big things come in small packages." In *Proceedings of the 24th International Conference on World Wide Web Companion*, 1551–1552, International World Wide Web Conferences Steering Committee. 72

Russell, S. J. (1998), "Learning Agents for Uncertain Environments (Extended Abstract)." In *COLT* (P. L. Bartlett and Y. Mansour, eds.), 101–103, ACM. 99

Sandhaus, E. (2008), "The New York Times Annotated Corpus." *Linguistic Data Consortium, Philadelphia*, 6. 28

Shen, W., J. Wang, P. Luo, and M. Wang (2012), "LIEGE: Link Entities in Web Lists with Knowledge Base." In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (Q. Yang, D. Agarwal, and J. Pei, eds.), 1424–1432, Beijing, China. 42

Shinzato, K. and K. Torisawa (2004), "Acquiring Hyponymy Relations from Web Documents." In *HLT-NAACL 2004: Main Proceedings* (D. M. Susan Dumais and S. Roukos, eds.), 73–80, Association for Computational Linguistics, Boston, Massachusetts, USA. 129

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis (2016), "Mastering the Game of Go with Deep Neural Networks and Tree Search." *Nature*, 529, 484–489. 98

Singh, S., S. Riedel, B. Martin, J. Zheng, and A. McCallum (2013), "Joint Inference of Entities, Relations, and Coreference." In *Proceedings of AKBC*, 1–6, ACM. 96

Socher, R., C. C.-Y. Lin, A. Y. Ng, and C. D. Manning (2011), "Parsing Natural Scenes and Natural Language with Recursive Neural Networks." In *Proceedings of the 28th International Conference on Machine Learning (ICML)* (L. Getoor and T. Scheffer, eds.), 129–136, Omnipress, Bellevue, Washington, USA. 81

Sterckx, L., T. Demeester, J. Deleu, and C. Develder (2014), "Using Active Learning and Semantic Clustering for Noise Reduction in Distant Supervision." In *4th Workshop on Automated Base Construction at NIPS2014 (AKBC-2014)*, 1–6. 23, 39

Subramaniam, L. V., S. Roy, T. A. Faruquie, and S. Negi (2009), "A Survey of Types of Text Noise and Techniques to Handle Noisy Text." In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data* (D. P. Lopresti, S. Roy, K. U. Schulz, and L. V. Subramaniam, eds.), AND '09, 115–122, ACM, New York, NY, USA. 40, 69

Suchanek, F. M., G. Kasneci, and G. Weikum (2008), "YAGO: A Large Ontology from Wikipedia and WordNet." *Web Semantics: Science, Services and Agents on the World Wide Web*, 6, 203–217. 13, 21

Surdeanu, M. and H. Ji (2014), "Overview of the English Slot Filling Track at the TAC2014 Knowledge Base Population Evaluation." In *Proceedings of the TAC-KBP 2014 Workshop*. 2, 3, 10, 11, 12, 38, 120

Surdeanu, M., D. Mcclosky, J. Tibshirani, J. Bauer, A. X. Chang, V. I. Spitkovsky, and C. D. Manning (2010), "A Simple Distant Supervision Approach for the TAC-KBP Slot Filling Task." In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, NIST, Maryland, USA. 22

Surdeanu, M., J. Tibshirani, R. Nallapati, and C. D. Manning (2012), "Multi-instance Multi-label Learning for Relation Extraction." In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (J. Tsujii, J. Henderson, and M. Pasça, eds.), 455–465, Association for Computational Linguistics, Jeju Island, Korea. 21, 24, 25, 26, 53, 123

Sutton, C. and A. McCallum (2005), "Composition of Conditional Random Fields for Transfer Learning." In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing* (R. Mooney, C. Brew, P. C.-c. Lee-Feng Chien, Academia Sinica, and P. C.-c. Katrin Kirchhoff, University of Washington, eds.), 748–754, Association for Computational Linguistics, Vancouver, British Columbia, Canada. 86

Sutton, R. and A. Barto (1998), *Reinforcement Learning: An Introduction*. MIT Press. 97, 98

Swartz, A. (2002), "MusicBrainz: A Semantic Web Service." *IEEE Intelligent Systems*, 17, 76–77. 21

Takamatsu, S., I. Sato, and H. Nakagawa (2012), "Reducing Wrong Labels in Distant Supervision for Relation Extraction." In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (H. Li, C.-Y. Lin, M. Osborne, G. G. Lee, and J. C. Park, eds.), 721–729, Association for Computational Linguistics, Jeju Island, Korea. 22, 25, 39, 117

Tjong Kim Sang, E. F. and F. De Meulder (2003), "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition." In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003* (W. Daelemans and M. Osborne, eds.), 142–147, Edmonton, Canada. 4, 64, 72

Viswanathan, V., N. F. Rajani, Y. Bentor, and R. Mooney (2015), "Stacked Ensembles of Information Extractors for Knowledge-Base Population." In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (C. Zong and M. Strube, eds.), 177–187, Association for Computational Linguistics, Beijing, China. 43

Vlachos, A. (2012), "An Investigation of Imitation Learning Algorithms for Structured Prediction." In *Proceedings of the Tenth European Workshop on Reinforcement Learning (EWRL)* (M. P. Deisenroth, C. Szepesvári, and J. Peters, eds.), 143–154, Citeseer, Edinburgh, Scotland. 104, 127

Vlachos, A. and S. Clark (2014a), "A New Corpus and Imitation Learning Framework for Context-Dependent Semantic Parsing." *Transactions of the Association for Computational Linguistics*, 2, 547–559. 127

Vlachos, A. and S. Clark (2014b), "Application-Driven Relation Extraction with Limited Distant Supervision." In *Proceedings of the First AHA!-Workshop on Information Discovery in Text* (A. Akbik and L. Visengeriyeva, eds.), 1–6, Association for Computational Linguistics and Dublin City University, Dublin, Ireland. 22, 63, 65, 97, 115

Vlachos, A. and M. Craven (2011), "Search-based structured prediction applied to biomedical event extraction." In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning* (S. Goldwater and C. Manning, eds.), 49–57, Association for Computational Linguistics, Portland, Oregon, USA. 99, 127

Vrandečić, D. and M. Krötzsch (2014), "Wikidata: A Free Collaborative Knowledgebase." *Communications of the ACM*, 57, 78–85. 13, 21

Walker, C., S. Strassel, J. Medero, and K. Maeda (2006), "ACE 2005 multilingual training corpus." *Linguistic Data Consortium, Philadelphia*. 38, 64, 71, 73, 120

Wang, C., A. Kalyanpur, J. Fan, B. K. Boguraev, and D. Gondek (2012a), "Relation Extraction and Scoring in DeepQA." *IBM Journal of Research and Development*, 56, 9–1. 1

Wang, J., H. Wang, Z. Wang, and K. Q. Zhu (2012b), "Understanding Tables on the Web." In *ER* (P. Atzeni, D. W. Cheung, and S. Ram, eds.), volume 7532 of *Lecture Notes in Computer Science*, 141–155, Springer. 42

West, R., E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin (2014), "Knowledge Base Completion via Search-Based Question Answering." In *Proceedings of the 23rd International Conference on World Wide Web* (C.-W. Chung, A. Z. Broder, K. Shim, and T. Suel, eds.), 515–526, ACM, New York, NY, USA. 63, 65, 95, 115, 117, 130

Whitelaw, C., A. Kehlenbeck, N. Petrovic, and L. Ungar (2008), "Web-scale Named Entity Recognition." In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, 123–132, ACM, New York, NY, USA. 71

Wick, M., K. Rohanimanesh, A. Culotta, and A. Mccallum (2009), "Samplerank: Learning preference from atomic gradients." In *In Neural Information Processing Systems (NIPS), Workshop on Advances in Ranking*, 69–73. 24

Wu, D., W. S. Lee, N. Ye, and H. L. Chieu (2009), "Domain adaptive bootstrapping for named entity recognition." In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (P. Koehn and R. Mihalcea, eds.), 1523–1532, Association for Computational Linguistics, Singapore. 71

Wu, F. and D. S. Weld (2007), "Autonomously Semantifying Wikipedia." In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management* (M. J.

Silva, A. O. Falcão, A. H. F. Laender, R. Baeza-Yates, D. L. McGuinness, B. Olstad, and Ø. H. Olsen, eds.), 41–50, ACM, New York, NY, USA. 21

Wu, F. and D. S. Weld (2008), "Automatically Refining the Wikipedia Infobox Ontology." In *Proceedings of the 17th International Conference on World Wide Web* (J. Huai, R. Chen, H.-W. Hon, Y. Liu, W.-Y. Ma, A. Tomkins, and X. Zhang, eds.), WWW '08, 635–644, ACM, New York, NY, USA. 21, 22

Wu, F. and D. S. Weld (2010), "Open Information Extraction Using Wikipedia." In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (J. Hajič, S. Carberry, S. Clark, and J. Nivre, eds.), 118–127, Association for Computational Linguistics, Uppsala, Sweden. 117

Xu, W., R. Hoffmann, L. Zhao, and R. Grishman (2013), "Filling Knowledge Base Gaps for Distant Supervision of Relation Extraction." In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (P. Fung and M. Poesio, eds.), 665–670, Association for Computational Linguistics, Sofia, Bulgaria. 26, 39

Yang, B. and C. Cardie (2013), "Joint Inference for Fine-grained Opinion Extraction." In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (P. Fung and M. Poesi, eds.), 1640–1649, Association for Computational Linguistics, Sofia, Bulgaria. 96

Yao, L., S. Riedel, and A. McCallum (2010), "Collective Cross-document Relation Extraction Without Labelled Data." In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* (H. Li and L. M'arquez, eds.), 1013–1023, Association for Computational Linguistics, Cambridge, MA. 22, 24

Yao, X. and B. Van Durme (2014), "Information Extraction over Structured Data: Question Answering with Freebase." In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 956–966, Association for Computational Linguistics, Baltimore, Maryland. 1, 10

Yates, A., M. Banko, M. Broadhead, M. Cafarella, O. Etzioni, and S. Soderland (2007), "TextRunner: Open Information Extraction on the Web." In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics* (B. Carpenter, A. Stent, and J. D. Williams, eds.), 25–26, Association for Computational Linguistics, Rochester, New York, USA. 17, 18

Yu, X. and W. Lam (2010), "Jointly Identifying Entities and Extracting Relations in Encyclopedia Text via A Graphical Model Approach." In *Proceedings of COLING: Posters*, 1399–1407, Association for Computational Linguistics, Beijing, China. 96

Zhang, Z., A. L. Gentile, and I. Augenstein (2014), "Linked data as background knowledge for information extraction on the web." *ACM SIGWEB Newsletter*, 5. 7

Zhang, Z., A. L. Gentile, I. Augenstein, E. Blomqvist, and F. Ciravegna (2013a), "Mining Equivalent Relations from Linked Data." In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (P. Fung and M. Poesi, eds.), 289–293, Association for Computational Linguistics, Sofia, Bulgaria. 7

Zhang, Z., A. L. Gentile, I. Augenstein, E. Blomqvist, and F. Ciravegna (2015), "An Unsupervised Data-driven Method to Discover Equivalent Relations in Large Linked Datasets." *Semantic Web Journal*. To appear. 7

Zhang, Z., A. L. Gentile, E. Blomqvist, I. Augenstein, and F. Ciravegna (2013b), "Statistical Knowledge Patterns: Identifying Synonymous Relations in Large Linked Datasets." In *The Semantic Web - 12th International Semantic Web Conference (ISWC)* (H. Alani, L. Kagal, A. Fokoue, P. T. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. F. Noy, C. Welty, and K. Janowicz, eds.), volume 1, 703–719, Springer Berlin Heidelberg, Sydney, NSW, Australia. 7

Zheng, Z., X. Si, F. Li, E. Y. Chang, and X. Zhu (2012), "Entity Disambiguation with Freebase." In *Web Intelligence*, 82–89, IEEE Computer Society. 1

Zhou, Z.-H. and X.-Y. Liu (2010), "On Multi-Class Cost-Sensitive Learning." *Computational Intelligence*, 26, 232–257. 128

Ziebart, B. D., A. L. Maas, J. A. Bagnell, and A. K. Dey (2008), "Maximum Entropy Inverse Reinforcement Learning." In *AAAI* (D. Fox and C. P. Gomes, eds.), 1433–1438, AAAI Press. 99