University of Sheffield

Multimodal Word Sense Translation

Chiraag Lala

Supervisor: Professor Lucia Specia

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy in Computer Science

in the

Department of Computer Science

March 7, 2021

Acknowledgement

First of all, I would like to thank my PhD supervisor, Professor Lucia Specia, for accepting me as her student and guiding me throughout my study. I am grateful to her because, like a caring mother of a difficult child, she patiently tolerated my tantrums and supported me at every step of my PhD journey. Her discipline, work ethic and work-life-health balance are an inspiration. While I have never been able to match her qualities since the beginning of my PhD, I have also never stopped trying and aspiring to be like her. Besides the PhD, she has also helped me shape my career for which I am deeply indebted to her. Thank you so much :)

I would also like to thank my PhD co-supervisors, Dr Pranava Madhyastha and Dr Josiah Wang. Dr Madhyastha is a star when it comes to (a) Machine Learning and (b) dealing with pressures and deadlines. I have learnt a lot about these by listening to his discussions and by observing him work. Dr Wang and I share many common interests. He has been a great moral support and a friend in a supervisory role. I am grateful to have two amazing co-supervisors. Also, I thank Dr Mauricio Álvarez for reviewing my reports at various stages of the PhD and my father, Ramesh Lala, for helping me with proofreading this thesis :)

Next, I thank Dr Mareike Hartmann, Dr Stella Frank, Mr Charles Escudier, Dr Julia Ive, Dr Frederic Blain, Dr Pavel Pecina, Dr Jindřich Libovický, Mr Jindřich Helcl, and anyone else I might have missed, who helped to annotate and create the Multimodal Word Sense Translation Dataset which is at the heart of this PhD work :)

I thank many more people who may have had an indirect contribution to this thesis like the co-authors of my published papers, and the people with whom I might have discussed my PhD research, etcetera. To generalise even further, I thank everyone and everything in the past! That is because I believe in the 'butterfly effect' - a butterfly flapping its wings in the Amazon rainforest in the past can change the weather thousands of miles away in Siberia in the present or in the future. In other words, the present is the way it is because the past was the way it was. So I acknowledge everything in the past has contributed to this thesis in its current form. Pretty deep, right? :)

Finally, I thank my family (MaPaBhaJDZuDi) and friends (especially LxMLS People, MANiACs and MAC) for their love. This thesis is dedicated to Smile :)

Abstract

Artificial Intelligence research aims to bridge the gap between humans and computers. While humans are quite good at understanding images and text, machines still have a long way to go. Over the years, researchers have made significant progress in improving computers at 'processing' and 'understanding' images (Computer Vision) and text (Natural Language Processing). However, these advancements have remained mostly independent of each other. Only recently, researchers are beginning to merge the two disjointed fields of research by creating tasks like Image Captioning, Multimodal Machine Translation, Visual Question Answering, etcetera.

Multimodal Machine Translation is the task of translating text from one language to another given additional contextual information in other modalities like an image associated with the text. This thesis aims to study a specific problem encountered in Multimodal Machine Translation, which is the translation of ambiguous words. Translating an ambiguous word, which has multiple different meanings, is a challenge because depending on the context, its meaning (sense) could be different, and hence the translation could be different too. To study this specific problem of translating an ambiguous word given its multimodal contextual information, we propose a new task which we call Multimodal Word Sense Translation.

We created the dataset for the proposed task of Multimodal Word Sense Translation comprising of samples consisting an ambiguous word, its textual context (a sentence), its visual context (an image), and its correct translations conforming both the textual context and the visual context. Our dataset was created from Multi30K, an existing dataset for Multimodal Machine Translation, using word aligners to extract the words in the source language that get aligned to multiple different words in the target language followed by human filtering to clean the dataset further. Analysis of our dataset reveals the ambiguity of words can be of different types (textual ambiguity and visual ambiguity) and varying degree (some words are more ambiguous than others). One important use of our dataset is to evaluate Machine Translation models, both text-only and multimodal, at translating ambiguous words. So we also used our dataset to evaluate this particular aspect of the systems submitted to the Second and the Third Shared Task on Multimodal Machine Translation in the Conference on Machine Translation (WMT).

We developed several Machine Learning and Deep Learning models for the task of Multimodal Word Sense Translation. These include Bidirectional Long Short-Term Memory network that reads the textual context and the visual context as inputs and tags every ambiguous word in the sentence to its correct sense translation. We used our Multimodal Word Sense Translation models to re-rank the n-best translation outputs of a standard Seq2Seq Machine Translation model where we promote a lower-ranked translation output if it contains the correct sense translation of ambiguous words. This pipeline system was submitted to the Third Shared Task on Multimodal Machine Translation in the Conference on Machine Translation (WMT18). Our system was found to perform better than most other submissions in generating translation outputs with the correct sense translation of ambiguous words.

More experiments on Multimodal Word Sense Translation models were conducted with different data settings and different ways of integrating the textual context and the visual context to study their complementarities and their differences. Our findings reveal that the textual context is often more useful than the visual context for translating ambiguous words; however, for some cases, textual context alone is not sufficient, and the visual context is necessary. We also found the image representation commonly used by the research community for the visual context derived from an object detection model like ResNet may not be conducive for the task of Multimodal Word Sense Translation. So we propose a way to transform image representation to make it more favourable for our task using triplet loss. Another image representation that we found useful for our task is to use the objects detected in the image by an object detection system as word tokens and prepend or append them to the textual context. In addition to the Multimodal Word Sense Translation experiments, we used our model architectures in another similar task of Fill-in-the-blanks given multimodal contextual information.

Finally, we explored transfer learning for our task of Multimodal Word Sense Translation. More specifically, we studied the utility of pre-trained embeddings for our task. We found contextualised word embeddings like Bidirectional Encoder Representations from Transformers (BERT) and Embedding from Language Models (ELMo) to improve the performance of our models. However, contextualised joint vision and language embeddings like Visual-Linguistic Bidirectional Encoder Representation from Transformer (VL-BERT) do not seem to improve the performance further. We end the thesis with multimodal and multilingual transformer models for Multimodal Word Sense Translation.

In conclusion, in this thesis, we show Multimodal Word Sense Translation could benefit Multimodal Machine Translation and could potentially be useful in more tasks. We also found the visual context to improve the translation of ambiguous words but the improvements gained are minute mainly because visual ambiguities are fewer in our dataset.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine Translation, which refers to the translation of text from one language to another by a computer, is one of the first problems of Artificial Intelligence pioneered in the 1940s and the 1950s. It was publicly demonstrated for the first time in 1954 in the Georgetown-IBM experiment (Hutchins 1997) where more than sixty sentences in Russian were translated into English using a rule-based Machine Translation model comprising six 'grammar' rules. Since then, Machine Translation has grown in scale and has progressed away from rule-based approaches to corpus-based approaches to take advantage of existing corpora of translations. These approaches include Statistical Machine Translation (Koehn 2009) and Neural Machine Translation (Sutskever et al. 2014, Bahdanau et al. 2015, Vaswani et al. 2017) which learn translation patterns from a corpus of existing translations and use these to generate a translation of an unseen text.

In the early days of Machine Translation research, translation of words with multiple different meanings was identified as an important problem. Warren Weaver[1], in his memorandum on translation (Weaver 1949), looked at this problem of multiple meanings and formulated it as a separate task where the objective is to identify the correct meaning (sense[2]) of a given word in a given context from a list of pre-defined possible meanings (senses) of that word. Today this task is called Word Sense Disambiguation.

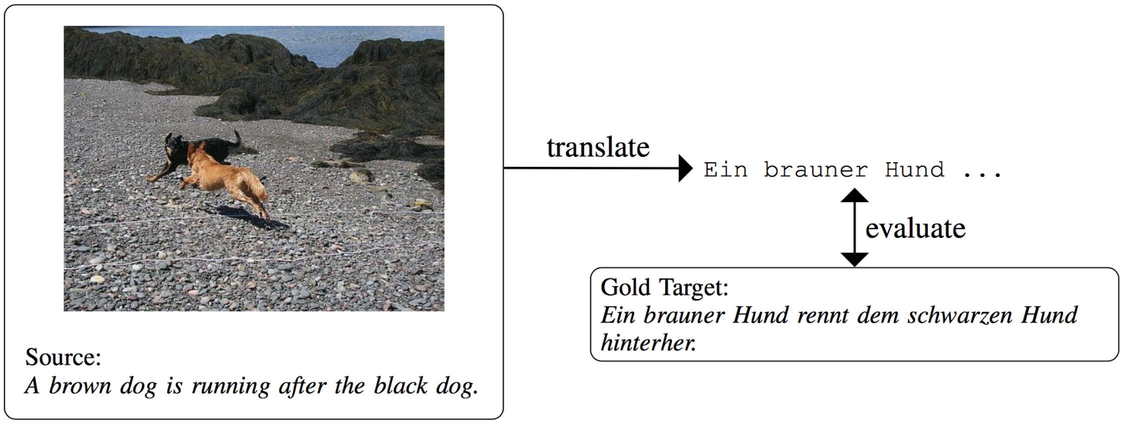
Weaver (1949) acknowledged the importance and the difficulty of Word Sense Disambiguation for Machine Translation. Bar-Hillel (1964) argued that Word Sense Disambiguation requires real-world knowledge making it extremely difficult for computers to solve. This inherent difficulty was the central point in Bar-Hillel's treatise on Machine Translation (Bar-Hillel 1964) in which he asserted that he saw no means by which the sense of the word *pen* in the sentence "The box is in the pen" could be determined automatically. This led to the Automatic Language Processing Advisory Committee report (ALPAC 1966), which is generally regarded as the direct cause for the abandonment of most research on machine translation

[1] An American scientist and mathematician who is widely recognized as one of the pioneers of Machine Translation.

[2] The notion of 'sense' was introduced in Frege (1892) referring to the specific meaning of a word in a given context. A word can have multiple senses depending on its context and usage like the word 'second' could refer to the unit of time or the number 2 position. Multiple different words referring to the same object may also have different senses like the words 'cat', 'kitty', 'mouser', 'feline' which refer to the same animal but may have different senses. 'Kitty' may refer to a baby cat and 'mouser' may refer to a fully grown wild cat that catches a mouse. In this thesis, we use 'meaning' of a word and its 'sense' interchangeably.

in the early 1960s. So, over time, researchers began working on Word Sense Disambiguation separately and largely independently from Machine Translation as a monolingual (one language only, mostly English) and a monomodal (text-only) task. Many approaches for Word Sense Disambiguation ranging from rule-based to supervised learning have been developed and explored over the years (Agirre & Edmonds 2007, Navigli 2009, Raganato et al. 2017). These systems are trained on large sense-tagged corpora annotated by humans with sense tags from a pre-defined sense inventory such as WordNet (Fellbaum 2012).

While significant progress has been made over the years in both Machine Translation and Word Sense Disambiguation separately, these have primarily remained monomodal (text-only) tasks where contextual information in other modalities like images have largely been ignored. This changed with Barnard & Johnson (2005), who used images as contextual information for Word Sense Disambiguation, and Calixto et al. (2012), who explored the creation of a dataset of images for Machine Translation. For translation, a new task called Multimodal Machine Translation was invented, which refers to translating text from one language to another by a computer using the information in other modalities as auxiliary cues. It has been recently framed as a shared task as part of the last three editions of the Conference on Machine Translation (WMT16, WMT17, WMT18) (Specia et al. 2016, Elliott et al. 2017, Barrault et al. 2018). Within the Conference on Machine Translation, the Multimodal Machine Translation task is defined as - given an image and its description in the source language, the objective is to translate the description into a target language, where this process can be supported by the information from the image, as depicted in Figure 1.1.



Ein brauner Hund ...

Gold Target:
Ein brauner Hund rennt dem schwarzen Hund hinterher.

Source:
A brown dog is running after the black dog.

Figure 1.1: *Multimodal Machine Translation Shared Task*

One of the main motivations to introduce multimodality in Machine Translation is Word Sense Disambiguation. More specifically, it is the intuition that information from other modalities could help find the correct meaning (sense) of ambiguous[3] words in the source sentence, which could potentially lead to more accurate lexical choices of those words in the translation. For example, the English sentence “A man is holding a seal” could have at least two different translations in German depending on the meaning (sense) of the word

[3] Words with multiple different meanings or senses.

seal. These could be (1) “Ein Mann hält ein Siegel” where ‘Siegel’ refers to a ‘stamp’ seal in German, or (2) “Ein Mann hält einen Seehund” where ‘Seehund’ refers to the ‘animal’ seal in German. The images (Figures 1.2 and 1.3) could help a Multimodal Machine Translation system disambiguate the correct sense of the word *seal* and translate accordingly.

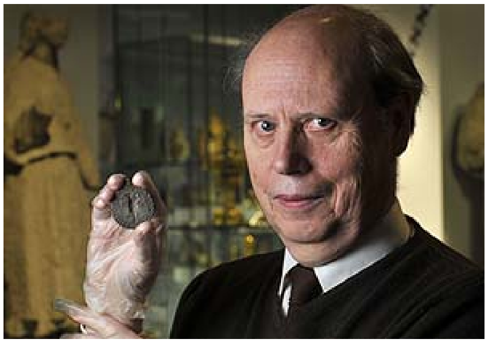


Figure 1.2: *“A man is holding a seal” → “Ein Mann hält ein Siegel”. Translation depending on the ‘stamp’ sense of the word ‘seal’ from the image.*

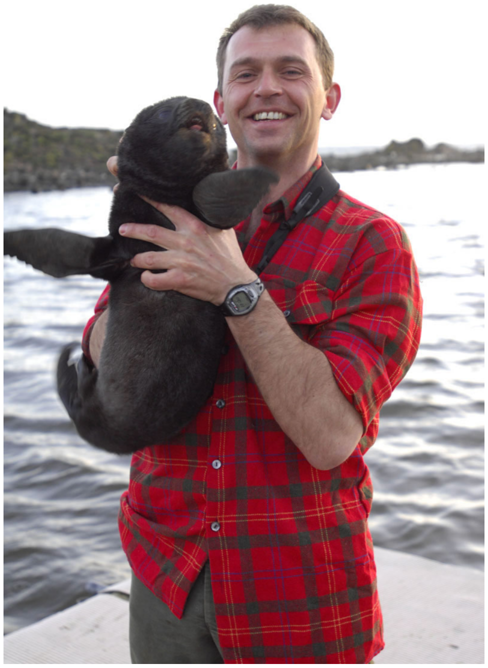


Figure 1.3: *“A man is holding a seal” → “Ein Mann hält einen Seehund”. Translation depending on the ‘animal’ sense of the word ‘seal’ from the image.*

In standard monomodal Word Sense Disambiguation, words are disambiguated based only on their textual context. However, in a multimodal setting, we could also disambiguate words using visual context. This modified version of Word Sense Disambiguation that uses visual context instead of textual context is called Visual Sense Disambiguation. In monolingual work, Visual Sense Disambiguation has previously been attempted for ambiguous nouns like the word 'bank' which could refer to a financial institution or a bank of a river (Barnard & Johnson 2005, Loeff et al. 2006, Saenko & Darrell 2009, Chen et al. 2015). Recently, Visual Sense Disambiguation has also been attempted for ambiguous verbs like the word 'play' which could refer to playing a musical instrument or playing a sport (Gella et al. 2016).

In Machine Translation, including Multimodal Machine Translation, Word Sense Disambiguation happens implicitly. For instance, in the same example, "A man is holding a seal", we would come to know whether a Machine Translation system disambiguated the correct sense of the word *seal* only indirectly from the translation produced by the system. The corresponding translation of the word *seal* in the target language (*Siegel* or *Seehund* in German) acts as a "sense label". This is the same as the cross-lingual form of Word Sense Disambiguation which was introduced in Resnik & Yarowsky (1999) and explored in the Semantic Evaluation (SemEval) shared tasks (Lefever & Hoste 2010, 2013). In this task, the objective is to identify the correct sense label of an ambiguous word given its textual context where the sense label is derived from the translation of the ambiguous word into a different language.

Cross-lingual Word Sense Disambiguation is a monomodal task where only textual context is available for disambiguating the word sense. It had not been explored in a multimodal setting where contextual information in multiple modalities can be used to disambiguate ambiguous words. Also, the emphasis of this task has been Word Sense Disambiguation and not Translation of ambiguous words or evaluation and improvement of Machine Translation systems at translating ambiguous words. Further, in Multimodal Machine Translation, we would like to know which modality (visual or textual) contributed to the correct or incorrect translation of ambiguous words and to what extent. After all, one of the motivations to introduce multimodality in Machine Translation is additional modality will help translate ambiguous words correctly. Therefore, the focus of this PhD is to study this specific property of translating ambiguous words and preserving their meanings (senses) in the translation given multimodal contextual information. We call it Multimodal Word Sense Translation.

1.1 Aims and Objectives

The aims and objectives of this thesis are the following:

1. Create a dataset to study Multimodal Word Sense Translation. More specifically,
   (a) The dataset should have ambiguous words, each of which have multiple different lexical translations with different meanings (senses).
   (b) Each sample of an ambiguous word must be accompanied by contextual information in multiple modalities, a text and an image.
   (c) The lexical translation of the ambiguous word in a sample should conform to the multimodal contextual information, both text and image, preserving the meaning (sense) of the ambiguous word.
2. Evaluate Multimodal Machine Translation systems at translating ambiguous words. More specifically,
   (a) Evaluate the systems submitted to the Multimodal Machine Translation shared task in the Conference on Machine Translation (WMT).

   (b) Compare with human evaluation and other automatic metrics to evaluate Machine Translation systems.
   (c) Investigate the utility of additional modality in Multimodal Machine Translation systems for translating ambiguous words.

3. Develop and explore models for Multimodal Word Sense Translation. More specifically,
   (a) Develop models for Monomodal Word Sense Translation and Multimodal Word Sense Translation to study the utility and effectiveness of contextual information in different modalities for translating ambiguous words.
   (b) Utilize these models to improve a standard Machine Translation system. In other words, incorporate a Multimodal Word Sense Translation model into a Machine Translation system.

4. Investigate pre-trained embeddings for Multimodal Word Sense Translation. More specifically, explore the utility of the following pre-trained representations:
   (a) Pre-trained word embeddings like Word2Vec (Mikolov, Sutskever, Chen, Corrado & Dean 2013) and Global Vectors for word representation (GloVe) (Pennington et al. 2014)
   (b) Pre-trained contextualised word embeddings like Embeddings from Language Models (ELMo) (Peters et al. 2018) and Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2018)
   (c) Pre-trained Multimodal contextualised word embedding like Visual-Linguistic Bidirectional Encoder Representation from Transformer (VL-BERT) (Su et al. 2020)
   (d) Visual features extracted from the pre-trained 50-layer Residual Network (ResNet-50) object recognition model (He et al. 2016).

1.2 Contributions

The main contributions of this thesis are the following:

1. We introduce an exhaustive[4] approach to create a dataset for Multimodal Word Sense Translation from a dataset for Multimodal Machine Transaltion using word aligners. With our approach we created the Multimodal Word Sense Translation Dataset[5] (Lala & Specia 2018) from the Multi30K dataset (Elliott et al. 2016). Our dataset consists of more than 1,100 unique ambiguous words in English with multiple different lexical translations of different senses in German or French or Czech. For these ambiguous words, we extracted more than 100,000 samples in total consisting of an ambiguous word and its word sense translation together with contextual information in multiple modalities, which is a sentence and an image, to identify the sense translation.

[4] Our approach begins with all possible words in the source language that have multiple different lexical translations in the target language in a given parallel corpus as candidates to be ambiguous words. We then filter out the unambiguous instances. This is a top-down 'exhaustive' approach where we start with more candidate words and then filter out unambiguous ones as against a bottom-up approach where we start with an empty list and add samples of ambiguous words to it.

[5] Previously known as Multimodal Lexical Translation Dataset: `https://github.com/sheffieldnlp/mlt`

2. We evaluated systems submitted to the Multimodal Machine Translation shared task of 2017 and 2018 in the Conference on Machine Translation (WMT17, WMT18). We checked if these systems translated ambiguous words correctly using our dataset. It was called the 'Lexical Translation Accuracy' (LTA) metric in Barrault et al. (2018). We also evaluated various other systems in Specia et al. (2020). The Lexical Translation Accuracy metric was also used in several other research works in the Multimodal Machine Translation domain.

3. We developed and experimented with several Machine Learning and Deep Learning models for the task of Multimodal Word Sense Translation. This includes developing and experimenting with Bidirectional Long Short-Term Memory (BLSTM) network[6] (Hochreiter & Schmidhuber 1997, Graves & Schmidhuber 2005) that reads the textual context and the visual context as inputs and tags every ambiguous word in the sentence to its correct sense translation (Lala et al. 2019).

4. We developed a Multimodal Machine Translation pipeline system which uses a Multimodal Word Sense Translation model (Lala et al. 2018). In our approach, the Multimodal Word Sense Translation model is used to re-rank the n-best translation outputs of a standard Seq2Seq Machine Translation model where we promote a lower-ranked translation output if it contains the correct sense translation of ambiguous words in the input text given the image. Our re-ranking approach was inspired by our experiments in Lala et al. (2017) where we studied the potential scope of re-ranking n-best translation outputs via an 'Oracle' experiment. This pipeline system was submitted to the Third Shared Task on Multimodal Machine Translation in the Conference on Machine Translation (WMT18) (Barrault et al. 2018). It performed better than most other submissions in generating translation outputs with the correct sense translation of ambiguous words as measured by the Lexical Translation Accuracy metric.

5. We carried out a detailed analysis of our dataset which includes a human annotation experiment to investigate the nature of ambiguous words and the utility of the visual context and the textual context for Word Sense Translation. We also conducted several experiments with different model architectures, data settings, and pre-trained representations to probe the utility of contextual information in different modalities for Multimodal Word Sense Translation. Our findings for Multimodal Word Sense Translation are similar to those for Multimodal Machine Translation as found in Caglayan et al. (2019).

1.3 Published Work

During my PhD tenure, I contributed to the following list of papers which were published in reputed international conferences and scientific journals.

1. Lala et al. (2017): Lala, Chiraag, Pranava Madhyastha, Josiah Wang, and Lucia Specia. "Unraveling the Contribution of Image Captioning and Neural Machine Translation for Multimodal Machine Translation." The Prague Bulletin of Mathematical Linguistics 108, no. 1 (EAMT). 2017.

[6] https://github.com/ImperialNLP/mltcode

2. Lala & Specia (2018): Lala, Chiraag, and Lucia Specia. "Multimodal lexical translation." In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC). 2018.

3. Lala et al. (2018): Lala, Chiraag, Pranava Swaroop Madhyastha, Carolina Scarton, and Lucia Specia. "Sheffield Submissions for WMT18 Multimodal Translation Shared Task." In Proceedings of the Third Conference on Machine Translation (WMT): Shared Task Papers. 2018.

4. Lala et al. (2019): Lala, Chiraag, Pranava Swaroop Madhyastha, and Lucia Specia. "Grounded Word Sense Translation." In Proceedings of the Second Workshop on Shortcomings in Vision and Language (NAACL). 2019.

1.4 Overview of the thesis

- In Chapter 2 we present the relevant **background and literature review** which sets up the foundation for the task of Multimodal Word Sense Translation. We begin with definitions of Machine Translation and Words Sense Disambiguation covering the seminal works in the two fields of research. Then we explore works which integrate the two fields. Until now, only monomodal works on Machine Translation and Word Sense Disambiguation are explored. Next, we review the various language and vision tasks which gained popularity, especially in the last decade promoting multimodal machine learning. We then review research papers in multimodal versions of Machine Translation and Word Sense Disambiguation which are Multimodal Machine Translation and Visual Sense Disambiguation respectively.

- In Chapter 3 we present our approach of creating the **Dataset for Multimodal Word Sense Translation**. We begin with a literature review of datasets for Multimodal Machine Translation, Visual Sense Disambiguation, and evaluation metrics for these tasks. Then we present the methodology we adopted to create our dataset. Exploratory data analysis and a human annotation experiment to investigate the nature of ambiguous words in our dataset and the utility of the visual context and the textual context for Word Sense Translation by a human are also presented. Use of our dataset to evaluate Multimodal Machine Translation systems at translating ambiguous words and comparing it with other evaluation metrics is also discussed.

- In Chapter 4 we develop and explore **Models for Multimodal Word Sense Translation**. We begin with a literature review on different models for relevant tasks like Word Sense Disambiguation and describe the model architectures we develop for our task of Multimodal Word Sense Translation. Experiments with different data settings are also explored followed by an analysis of the results. Use of our models to re-rank the n-best translation outputs of a standard Machine Translation system is presented. Use of our model architecture for other similar tasks like 'Fill-in-the-blanks using multimodal contextual information' is also mentioned.

- In Chapter 5 we explore **Image Features and Word Embeddings for Multimodal Word Sense Translation**. We begin with a literature review of relevant research work on pre-trained word embeddings, image representations, and joint vision and language (multimodal) representations. Then we present our experiments with some of these pre-trained word embeddings and image representations and evaluate their usefulness for our task of Multimodal Word Sense Translation. A way to transform image representations to be more conducive for Multimodal Word Sense Translation via triplet loss is also described.

- Finally, in Chapter 6 we summarise all our findings from the previous chapters and **conclude** the thesis. We also discuss the future directions of Multimodal Word Sense Translation research.

Chapter 2

Background and Literature Review

This PhD thesis on Multimodal Word Sense Translation is inspired by the tasks of (a) Machine Translation, (b) Word Sense Disambiguation and their multimodal extensions, (c) Multimodal Machine Translation, and, (d) Visual Sense Disambiguation respectively. Extension of monomodal (text-only) Natural Language Processing tasks to their multimodal[1] (text and vision) versions is a recent trend driven by advances in Computer Vision and Multimodal Machine Learning in an effort to emulate humans who can process, integrate and use information in multiple modalities to solve various complex tasks. In this chapter, we will elaborate on these topics and cover the relevant background and literature to set the foundation for Multimodal Word Sense Translation.

We begin by defining and formalising **Machine Translation** in section 2.1. We will cover the two prominent paradigms of Machine Translation which are Statistical Machine Translation and Neural Machine Translation. We will also look at the major errors in Machine Translation which include 'inaccurate lexical choice' in the translation due to 'lexical ambiguities' like 'category ambiguity', 'homography and polysemy', and 'transfer ambiguity'.

In section 2.2, we define and formalise **Word Sense Disambiguation** and explore the different approaches adopted for this task. We will also look at Cross-lingual Word Sense Disambiguation, where the sense labels are lexical translations of the ambiguous words, which is most relevant to Multimodal Word Sense Translation.

Next, we explore **Word Sense Disambiguation in Machine Translation** in section 2.3. We explore how Word Sense Disambiguation was incorporated into Statistical Machine Translation systems and also the approaches which incorporate it in Neural Machine Translation. We will also cover how Word Sense Disambiguation capabilities of Neural Machine Translation were evaluated.

In section 2.4, we discuss some of the **Language and Vision Tasks** that have been recently introduced. These multimodal extensions of Natural Language Processing tasks include Multimodal Machine Translation and Visual Sense Disambiguation which are explored in extensive detail in sections 2.4.2 and 2.4.3 respectively. The success of Deep Convolutional Neural Networks for Image Classification which lead to the success of Image Captioning have set the stage for more interest in Language and Vision tasks. This is reviewed in section

[1] In general, 'multimodal' could refer to combination of many different modalities like textual, visual, auditory, tactile, etcetera. However, in this PhD thesis, we only consider the combination of textual and visual modalities as multimodal.

2.4.1 alongwith a discussion on image features extracted from Deep Convolutional Neural Networks for Language and Vision tasks.

2.1 Machine Translation

Machine Translation refers to automated translation of text carried out by a computer from one human language (for example English) to another (for example German). The first language is called the *source language* and the second language is called the *target language*. A sentence in the source language that needs to be translated is called the *source sentence* and its correct translation in the target language, usually decided by an expert human translator, is called the *reference translation*. We shall call a pair of a source sentence and its reference translation a *parallel pair*. A collection of parallel pairs is called a *parallel corpus*.

Early approaches to Machine Translation were largely inspired by the study of linguistics and made use of morphological, syntactic and semantic regularities of the source language and the target language. The regularities and patterns were retrieved from monolingual dictionaries, bilingual dictionaries, and, grammars of the languages that were prepared by linguists to devise rules for automatic translation. Detailed descriptions of many such rule-based Machine Translation approaches can be found in Hutchins & Somers (1992). As rule-based approaches were being tried out, vast amounts of human translated parallel corpora were also being generated that were becoming available like the Canadian Hansards parliamentary proceedings in English and French (Roukos et al. 1995). It was realized that such datasets of example translations could be used to train Machine Translation systems without explicitly programming the rules for translation (Brown et al. 1988, Och 2002).

A Machine Translation task is often posed and approached at the level of sentences and can be then extended to the level of documents. At the level of sentences, the objective of a Machine Translation system is to generate a translation of the source sentence which is either identical or semantically similar[2] to the reference translation. Formally, at the level of sentences, if x is a source sentence and Y is the set of all possible sentences in the target language, then a probability distribution $p(y|x)$, where $y \in Y$, could be used to find the best translation $\hat{y}$ by selecting the translation hypothesis that has the highest conditional probability in the distribution.

$$\hat{y} = \arg\max_{y \in Y} p(y|x) \tag{2.1}$$

However, modelling the probability distribution $p(y|x)$ of translations for a given source sentence is a challenge because of several reasons like the set Y of all translations is infinitely large and there is no known function that can measure semantic similarity of two sentences. And we have to also contend with the ambiguities and complexities of human languages. To model the probability distribution $p(y|x)$, two distinct strategies have emerged. These are Statistical Machine Translation and Neural Machine Translation.

[2] Two sentences are said to be semantically similar if they have the same meaning even if they are ‘lexically’ different (words are different) or ‘syntactically’ different (word order is different) or both.

2.1.1 Statistical Machine Translation

In Statistical Machine Translation, the problem of modeling the probability distribution $p(y|x)$ of translations for a given source sentence (Equation 2.1) is approached using 'noisy channel' (Shannon et al. 1949) by applying Bayes' Theorem (Appendix A.1) as follows:

$$\hat{y} = \arg\max_{y \in Y} p(y|x) = \arg\max_{y \in Y} \frac{p(x|y)p(y)}{p(x)} = \arg\max_{y \in Y} p(x|y)p(y) \tag{2.2}$$

The probability distribution $p(y)$ of observing a sentence y in the target language is called the *language model* of the target language. The probability distribution $p(x|y)$ of observing a sentence x in the source language given y as its translation in the target language is called the *translation model*. The probability $p(x)$ of observing a sentence x in the source language, which is the language model of the source language, is same for all $y \in Y$. Mathematically, this does not change the most probable translation $\hat{y}$ of the source sentence x, so we don't need to compute it in equation 2.2.

The language model $p(y)$ and the translation model $p(x|y)$ are estimated further by making an assumption that occurrence of words in a sentence are random events and then using the chain rule (Appendix A.3) on these random events. If sentence y in the target language is a sequence of n_y words $(y_1, y_2, ..., y_{n_y})$ and the source sentence x is a sequence of n_x words $(x_1, x_2, ..., x_{n_x})$ then the language model and the translation model are estimated as follows:

$$p(y) = p(y_{n_y}, y_{n_y-1}, y_{n_y-2}, ..., y_1) = p(y_1) \cdot \prod_{k=2}^{n_y} p(y_k|y_{k-1}, y_{k-2}, ..., y_1) \tag{2.3}$$

$$p(x|y) = p(x_{n_x}, x_{n_x-1}, x_{n_x-2}, ..., x_1|y) = p(x_1|y) \cdot \prod_{k=2}^{n_x} p(x_k|x_{k-1}, x_{k-2}, ..., x_1, y) \tag{2.4}$$

More assumptions can be made to simplify the conditional probabilities further and then computed from the statistics of the datasets. For example, the n-gram language model assumes the occurrence of a word depends only on the $n-1$ previous words. So we compute the conditional probability terms as follows,

$$p(y_k|y_{k-1}, ..., y_1) \approx p(y_k|y_{k-1}, ..., y_{k-n+1}) \approx \frac{count(y_k, y_{k-1}, ..., y_{k-n+1})}{count(y_{k-1}, ..., y_{k-n+1})} \tag{2.5}$$

where $count(z)$ refers to counting the occurrence of z in the dataset. In general, the language model and the translation model can be estimated and computed from the statistics of parallel corpora and monolingual datasets. Also, by making certain assumptions, more models covering different aspects of translation can be computed independently and added to the equation 2.2 as follows,

$$\hat{y} = \arg\max_{y \in Y} p_1(x, y)^{\lambda_1} \cdot p_2(x, y)^{\lambda_2} \cdots p_m(x, y)^{\lambda_m} \tag{2.6}$$

where m different models weighted by λ_i parameters are being multiplied. This is referred to as the 'noisy channel'. The parameters λ_i are learned via Minimum Error Rate Training (Och 2003). The mathematics of the noisy channel modelling of Statistical Machine Translation is described in Brown et al. (1993). Most of the research in Statistical Machine Translation has been about feature engineering and adding new models to the noisy channel (equation 2.6).

The early Statistical Machine Translation methods were primarily *word-based* where individual words are treated as atomic units of translation. Koehn et al. (2003) proposed a *phrase-based* approach to Statistical Machine Translation where phrases[3] are treated as atomic units of translation. This resulted in impressive improvements. First, the source sentence is segmented into phrase units. Each of the units is translated into a target language unit and then reordered as depicted in figure 2.1.

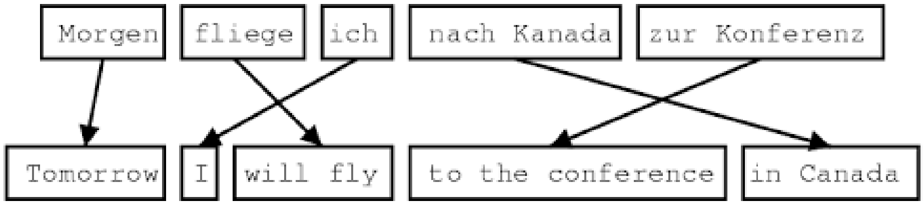


Figure 2.1: *An Example of Phrase-based Translation. Figure taken from Koehn (2009).*

At the core of a phrase-based Statistical Machine Translation system is the *Phrase Translation Table* which is a lexicon of phrases in the source and the target languages that translate into each other, with a probability distribution. More formally, let X_p and Y_p be the sets of all phrases in the source and the target languages respectively, then a database of conditional probabilities $p(x_p|y_p)$, $\forall(x_p, y_p) \in X_p \times Y_p$, is called the phrase translation table. These conditional probabilities $p(x_p|y_p)$ from the phrase translation table are then used in estimating the translation model $p(x|y)$ in equation 2.4.

The mathematical formulation of the noisy channel in equation 2.6 and the Phrase Translation Table are the basic ideas of the current state-of-the-art Statistical Machine Translation systems like Moses (Koehn et al. 2007). More details on other aspects of Statistical Machine Translation like word alignments, decoding algorithm with beam search, etcetera can be found in Koehn (2009). Recently, the noisy channel approach of modelling $p(y|x)$ has been challenged by a more direct approach using Neural Networks.

2.1.2 Neural Machine Translation

Neural Machine Translation is a new paradigm of Machine Translation where translations are obtained using a Neural Network[4] whose weights / parameters are derived from the parallel corpus using the backpropagation algorithm (Rumelhart et al. 1986, Werbos 1990). It is a radical departure from Statistical Machine Translation because Statistical Machine Translation consists of subcomponents like language model, translation model, distortion model, etcetera as shown in equation 2.6 that are separately engineered, while in Neural Machine Translation the probability distribution $p(y|x)$ in equation 2.1 is modelled directly using novel neural network architectures like *Recurrent Neural Network Encoder-Decoder* (Sutskever et al.

[3] Not to be confused as the linguistic definition of a phrase. Here, any contiguous sequence of words is called a phrase.

[4] Neural Network is an information processing paradigm that is loosely inspired by the way biological nervous system processes information. A basic introduction to Neural Networks is given in Chapter 6 in Goodfellow et al. (2016)

2014), *Attention-based Recurrent Neural Network Encoder-Decoder* (Bahdanau et al. 2015) and *Transformer* (Vaswani et al. 2017), where all parts of the model are trained jointly (end-to-end).

The concept of Recurrent Neural Network in its simplest form was introduced in Elman (1990) where the network maintains a hidden state vector h_t at every time step t. This hidden state vector h_t is computed from the previous hidden state vector h_{t-1} of the previous time step and the input vector x_t of the current time step. It is formulated as follows:

$$h_t = \sigma_h(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \tag{2.7}$$

where σ_h is some activation function (Appendix A.4), W_{hx} and W_{hh} are matrices which are the weights / parameters of the network, and b_h is the bias vector which is another parameter of the network. The output vectors y_t can be obtained from the hidden state vectors h_t as follows:

$$y_t = \sigma_y(W_y h_t + b_y) \tag{2.8}$$

where σ_y is some activation function (Appendix A.4), W_y is a matrix of weights / parameter, and b_y is a bias vector parameter. The Recurrent Neural Network can be seen as a language model where hidden state vector h_t is modelling $p(x_1, x_2, ..., x_t)$ of the input sequence of t words and output vector y_t is modelling the conditional probability $p(y_t|x_1, x_2, ..., x_t)$. More sophisticated Recurrent Neural Networks with complex formulations that are able to model the input sequence better were developed later like Long Short-Term Memory networks (Hochreiter & Schmidhuber 1997) and Gated Recurrent Unit (Chung et al. 2014).

Sutskever et al. (2014) demonstrated the use of Recurrent Neural Networks for Machine Translation. More specifically, two Long Short-Term Memory networks called the Encoder and the Decoder were used. The Encoder learns to encode the input sentence and the Decoder learns to decode the encoder representation to generate a translation as depicted in figure 2.2. From a probabilistic perspective, the Decoder part of these architectures at time step t can be seen as modelling $p(y_t|x, y_1, y_2, ..., y_{t-1})$. When seen as a whole, it fits well to modelling the Machine Translation problem in equation 2.1.

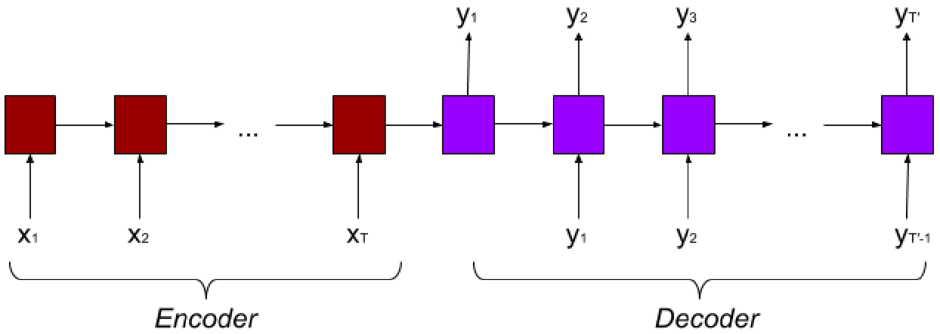


Figure 2.2: *Encoder-Decoder Architecture using two Recurrent Neural Networks.*

The Recurrent Neural Network Encoder-Decoder model shown above in Figure 2.2 is trained jointly (end-to-end) using the *backpropagation through time* algorithm (Werbos 1990). It performs very well for short sentences but the performance degrades for longer sentences (Cho et al. 2014, Toral & Sánchez-Cartagena 2017) suggesting the encoder representation used in the decoder by such models, just the final hidden state vector h_T of the Encoder, lacks representational power. To address this, Bahdanau et al. (2015) proposed using all the hidden states of the Encoder at every time step of the Decoder with what is called the 'attention mechanism'.

Formally the attention mechanism is described as follows: let h_j^{enc} be the hidden state vector of the Encoder for the j^{th} input time-step corresponding to x_j and let h_k^{dec} be the hidden state vector of the Decoder for the k^{th} output time step corresponding to the output word y_k. In attention mechanism, a context vector c_k is computed as a weighted sum of all Encoder hidden states as follows:

$$c_k = \sum_{j=1}^{T} \alpha_{kj} \cdot h_j^{enc} \tag{2.9}$$

where attention weights α_{kj} are computed from an alignment model a which computes alignments between Encoder hidden states and Decoder hidden states as follows:

$$\alpha_{kj} = \frac{\exp(a(h_{k-1}^{dec}, h_j^{enc})}{\sum_{l=1}^{T} \exp(a(h_{k-1}^{dec}, h_l^{enc})} \tag{2.10}$$

Finally, the Decoder uses this context vector c_k in addition to previous decoder hidden state h_{k-1}^{dec} and previous output y_{k-1} to compute the current hidden state h_k^{dec} as follows:

$$h_k^{dec} = \sigma_{h^{dec}}(W_1 \cdot h_{k-1}^{dec} + W_2 \cdot y_{k-1} + W_3 \cdot c_k + b_{h^{dec}}) \tag{2.11}$$

After Bahdanau et al. (2015), more improvements happened when many layers of Recurrent Neural Network units were stacked to form deeper Attention-based Recurrent Neural Network Encoder-Decoder models (Luong et al. 2015, Wu, Schuster, Chen, Le, Norouzi, Macherey, Krikun, Cao, Gao, Macherey, Klingner, Shah, Johnson, Liu, Kaiser, Gouws, Kato, Kudo, Kazawa, Stevens, Kurian, Patil, Wang, Young, Smith, Riesa, Rudnick, Vinyals, Corrado, Hughes & Dean 2016). Later, Vaswani et al. (2017) proposed replacing Recurrent Neural Network units with *self-attention* which is a further generalization of the attention mechanism described above in equations 2.9, 2.10 and 2.11. A neural network Encoder-Decoder architecture with many stacked layers of self-attention is called a *transformer*[5]. The transformer relies only on attention mechanism and unlike a Recurrent Neural Network which computes hidden states sequentially[6], the transformer computes all the hidden states in parallel, getting rid of the need of recurrence in processing the input source sentence. Transformer-based translation models have gained massive popularity in recent times which is evident in the Conference on Machine Translation (WMT18) shared task on News (Bojar et al. 2018). Most submissions to the shared task employ Transformer architecture (Raganato & Tiedemann 2018).

[5] For a more detailed description of transformer, besides Vaswani et al. (2017), please read `http://jalammar.github.io/illustrated-transformer/`

[6] A Recurrent Neural Network computes the hidden states sequentially. Therefore, it reads the input sequence sequentially and generates the output sequence sequentially.

2.1.3 Lexical errors in Machine Translation

Translations generated by a Machine Translation system are usually evaluated for the overall performance using an automated metric or a human evaluation. Common automatic metrics include BiLingual Evaluation Understudy (BLEU) (Papineni et al. 2002), Metric for Evaluation of Translation with Explicit ORdering (METEOR) (Denkowski & Lavie 2014), and Translation Edit Rate (TER) (Snover et al. 2006). Common human evaluations include manually ranking the translations by human annotators (Federmann 2012) and monolingual direct assessment of semantic similarity between Machine Translation output and the reference translation or bilingual direct assessment of semantic similarity between Machine Translation output and the source sentence (Graham et al. 2017). All these evaluation methods, both automatic and manual, give an overall score to a Machine Translation system which gives an indication of the general performance of the system but these do not provide any additional information. Going beyond general performance of Machine Translation systems, Bentivogli et al. (2016), Toral & Sánchez-Cartagena (2017) and Specia et al. (2017) carried out a detailed analysis of errors made by phrase based Statistical Machine Translation systems and Neural Machine Translation systems. These specifc errors include *fluency*, *reordering* and *lexical* errors among others.

Fluency refers to how "human-like" is the translation generated by a system. There is no direct way of measuring fluency except, perhaps, manually by direct assessment of fluency by humans. In practice, it is measured indirectly in terms of perplexity (Appendix A.5) of the Machine Translation outputs on a Neural Language model. Formally, if w is a translation generated by a Machine Translation system consisting n words and $p(w)$ is the language model of the target language, then fluency (perplexity) is $p(w)^{-\frac{1}{n}}$.

Lexical translations of the words in a source sentence are often reordered in the translation because the syntax and grammar rules of a target language are usually different from those of a source language. Reordering error refers to the incorrect positioning of words in the translation. It is measured by checking for words in a Machine Translation output which are also found in the reference translation but get marked as a Word Error Rate (Popović & Ney 2011, Popović 2011) because they are in incorrect positions.

Other lexical errors include inflectional error, missing word error, extra word error, and lexical choice error. Inflectional error refers to words in the Machine Translation output which are not found in the reference translation but the base form of that word (its lemma or its stem)[7] is found in the base forms of the words in the reference translation. In other words, these are errors due to incorrect choice of the inflectional[8] form in the translation. Missing word error and extra word error, as the names suggest, are errors in the translation where a word is missing or an extra word is added. Lexical choice error refers to incorrect choice of words used in the Machine Translation output. For example, as mentioned in the introduction, if the source sentence is "A man is holding a seal" and if the reference translation is "Ein Mann hält einen Seehund" and then a Machine Translation system generates the

[7] Stemming and Lemmatization are methods to generate the base form, stem and lemma respectively, of inflected words. These differ because a stem may not be an actual word while lemma is a proper word. For example, given the word *studies* (and its other inflectional forms *study studying*, etcetera), its stem is *studi* which is not a proper word while its lemma is *study* which is a proper word.

[8] Inflection refers to different forms of a word which express a grammatical function or attribute such as tense, mood, person, number, case, and gender. For example, in terms of tense, the different inflections of the word *play* are *plays, played,*and *playing.*

translation as “Ein Mann hält ein Siegel” then the word ‘Siegel’ being chosen instead of ‘Seehund’ is considered a lexical choice error.

Fluency, reordering and lexical errors mentioned above are some of the main problems of Statistical Machine Translation. Neural Machine Translation systems have been shown to be more fluent with better reordering and more accurate generation of inflectional forms in the translation as compared to phrase-based Statistical Machine Translation; however, no improvements have been seen as far lexical choice errors are concerned (Bentivogli et al. 2016, Toral & Sánchez-Cartagena 2017). A similar result has been observed in Specia et al. (2017) where human annotators evaluated the quality of outputs from phrase-based Statistical Machine Translation and Neural Machine Translation systems. They actually observed an increase in lexical choice errors, which they called *mistranslation*, in Neural Machine Translation for English-Latvian language pair. Also, lexical choice errors (mistranslations) in Machine Translation comprise around 15% of all the different errors evaluated in Specia et al. (2017) and 47% of all the different errors evaluated in Vardaro et al. (2019) which is the most frequent of any error category. This suggests, the latest developments in Machine Translation have made little inroads in addressing inaccurate lexical choice (mistranslations) problem which is one of the biggest problems of Machine Translation. The main source of lexical choice errors (mistranslations) is the lexical ambiguities and is regarded as a “key bottleneck for progress in machine translation” (Dale et al. 2000, Tokowicz & Degani 2010, Djiako 2019).

Lexical ambiguities are of three kinds (Hutchins & Somers 1992) - (1) Category ambiguity, (2) Homography and Polysemy, and (3) Translation ambiguity. Category ambiguities are words which can have different grammatical or syntactic categories like the word *light* which can be a noun or a verb or an adjective depending on its usage in a sentence. Homographs and polysemes are words with multiple different meanings. Linguists identify homographs as those words which have multiple different meanings not related to each other. For example, the noun *bank* has two different unrelated meanings (financial institution or a side of a river). Polysemes are words which have multiple meanings but these are related in some way to each other like *branch* of a bank and *branch* of a tree are two different meanings of the word *branch* but related because one is a metaphorical extension of the other. Translation ambiguity refers to words which are not really ambiguous in the source language or not perceived as ambiguous by the native speakers of the source language but seen ambiguous from the perspective of the target language because it can be translated to multiple target language words or expressions like word *sportsperson* which does not have a gender in English but gets assigned a gender in the translation into French leading to different possible translations *sportive* (feminine) or *sportif* (masculine).

Resolving these lexical ambiguities is a task in itself, separate from Machine Translation, which is explored in section 2.2 below.

2.2 Word Sense Disambiguation

Resolving lexical ambiguities, also known as Word Sense Disambiguation, is the task of identifying the meaning of words using contextual information. It is an AI-complete[9] problem

[9] The most difficult problems of Artificial Intelligence are referred to as AI-complete problems. This terminology is inspired by NP-complete problems in Computational Complexity Theory.

(Mallery 1988) and has been historically conceived as the main task to be solved for effective machine translation. This is because of the the common intuition that disambiguation of ambiguous words in the source sentence should help translation systems make better lexical choices in the translation (Weaver 1949, Navigli 2009). A detailed account of the early history of Word Sense Disambiguation can be referred in Ide & Véronis (1998) and an introduction to modern approaches to the problem can be found in Navigli (2009).

In a standard Word Sense Disambiguation task, words are disambiguated based on their textual context. Formally, given a word w and its textual context T which is often the sentence in which w occurs, the objective of the task is to assign an appropriate sense to the word w. In other words, the objective is to learn a mapping f that maps a word (and its context) to a set of senses $S(w)$ of that word, such that $f(w,T) \in S(w)$. The set of senses $S(w)$ of a word w is encoded in a sense inventory, which is essentially a finite discrete set of meanings/senses for each word, or some discrete knowledge source like *WordNet* (Miller et al. 1990, Fellbaum 2012) and *BabelNet* (Navigli & Ponzetto 2012). Besides sense inventory, there are also sense-tagged corpora like SemCor corpus (Miller et al. 1994, Petrolito & Bond 2014) where samples of text are annotated with senses from WordNet. In practice, Word Sense Disambiguation is a classification task and evaluation of systems is usually performed in terms of classification Accuracy, Precision, Recall, and balance F-scores[10] (Appendix A.6).

2.2.1 Lesk and Supervised Approaches to Word Sense Disambiguation

A seminal approach to Word Sense Disambiguation is the Lesk algorithm (Lesk 1986) which compares the dictionary definition of the different senses of the ambiguous word with the textual context of the ambiguous word. In its simplest version, this comparison is measured by simply counting the number of common words. Formally, let w be an ambiguous word with a textual context T. Let $S(w) = \{s_1^w, s_2^w, ..., s_{|S|}^w\}$ be the set of different senses of w. Let $D(s_i^w)$ be the dictionary definition of sense s_i^w which is a text defining or explaining the sense. The key idea of the Lesk algorithm is to then define a relatedness function r which measures similarity (relatedness) between the textual contexts and the sense definitions. In the simplest version of the Lesk algorithm, $r(T, D(s_i^w))$ is total number of common words found in both T and $D(s_i^w)$ (More common words means they are more related or similar). Then the algorithm is to simply return the sense $\hat{s}$ with the highest similarity (relatedness) with the textual context T as follows:

$$f(w,T) = \hat{s} = \underset{s \in S(w)}{\arg\max}\, r(T, D(s)) \tag{2.12}$$

Over the years, there have been several modifications and extensions of the Lesk algorithm which have either modified the relatedness function r or modified the dictionary definitions of senses D. One such extension is the adapted Lesk algorithm (Banerjee & Pedersen 2002) inspired by the distributional hypothesis[11] (Harris 1968). In the adapted Lesk algorithm, first the dictionary definition of senses are obtained from WordNet (Fellbaum 2012) and then word vectors are created for every word in the WordNet corpus using word co-occurence counts. The word vectors of all words in a sense definition are added to obtain a vector representation

[10] A weighted harmonic mean of Precision and Recall.

[11] Words which are semantically similar are also distributionally similar, i.e. they have similar context or neighbouring words. In other words, "you shall know a word by the company it keeps" (Firth 1957).

of that sense. Similarly, word vectors of all words in the textual context are added to obtain a vector representation of the textual context. Finally, similarity (relatedness) r between sense definitions and the textual context is computed using cosine similarity (Appendix A.7) of their vector representations and this is used to obtain the correct sense of the ambiguos word using the formulation in equation 2.12. More variants of the Lesk algorithm are discussed and evaluated in Vasilescu et al. (2004), Agirre & Edmonds (2007). Several different vector representations of the senses and the textual context, and complex formulations of relatedness function have been explored over the years (Basile et al. 2014). The dictionary definitions or the sense inventories have also improved like the BabelNet (Navigli & Ponzetto 2012) which merges WordNet and Wikipedia.

The Lesk algorithm relies on sense definitions $D(s)$ which may not always be available. In some versions of Word Sense Disambiguation, sense inventory consists of only sense labels and no definitions or text related to that sense. In such cases, the training set consists of triples of the form (w, T, s) where w is an ambiguous word, T is its textual context and s is its sense label. Here, we can identify the correct sense $\hat{s}$ of w given T using a probability distribution $p(s|w, T)$ as follows:

$$f(w, T) = \hat{s} = \operatorname*{arg\,max}_{s \in S(w)} p(s|w, T) \tag{2.13}$$

Similar to Statistical Machine Translation, modelling of the conditional probability $p(s|w, T)$ can be done using the noisy channel approach (Shannon et al. 1949) using Bayes' Theorem (Appendix A.1). The noisy channel approach, similar to equation 2.6, to Word Sense Disambiguation is formulated as follows,

$$f(w, T) = \hat{s} = \operatorname*{arg\,max}_{s \in S(w)} p_1(w, T, s)^{\lambda_1} \cdot p_2(w, T, s)^{\lambda_2} \cdots p_m(w, T, s)^{\lambda_m} \tag{2.14}$$

where m different component models are engineered separately. The simplest form of the noisy channel approach is the Naive Bayes classifier (Webb 2010) which has been shown to perform very well for Word Sense Disambiguation (Brown et al. 1991, Mooney 1996, Escudero et al. 2000, Le & Shimazu 2004). Another classification algorithm that has been shown to perform very well for Word Sense Disambiguation is Support Vector Machines[12] (Lee & Ng 2002, Lee et al. 2004).

2.2.2 Neural Approaches to Word Sense Disambiguation

In recent times, there is a rising trend to use Neural Networks for Word Sense Disambiguation which allows a direct modelling of conditional probability $p(s|w, T)$ in equation 2.13. There have been several works which have demonstrated use of Recurrent Neural Networks for disambiguation (Yuan et al. 2016, Kågebäck & Salomonsson 2016, Raganato et al. 2017, Popov 2017). These approaches are nearly identical and differ mainly in how the Recurrent Neural Network reads the textual context T and whether it is disambiguating just one ambiguous word in T, which needs to be specified, or all the ambiguous words in T.

[12] Support Vector Machine is a supervised learning classification algorithm which seeks to create a hyper-plane or a set of hyper-planes in a high dimensional space. These hyper-plane(s) separate the different classes maximizing the margin between the hyperplane and the nearest data point of any class. For more detailed description of Support Vector Machines, refer Zhang (2010).

In Kågebäck & Salomonsson (2016), the textual context T which is a sentence containing the ambiguous word w is divided into two parts T_L and T_R. T_L (textual context to the left) is the sequence of words to the left of the ambiguous word in T. T_R (textual context to the right) is the sequence of words to the right of the ambiguous word in T. Two Long Short-Term Memory (LSTM) networks, one for T_L and one for T_R are trained to get semantic representations of the left and the right contexts respectively. These representations are then concatenated and used to obtain the sense label. A shortcoming of this approach is that we need to specify which word in the sentence needs disambiguation. Using it to disambiguate all words in the sentence will require us to iterate l number of times, where l is the sentence length.

Yuan et al. (2016) is a semi-supervised approach which uses one Long Short-Term Memory network. First, the ambiguous word w in the textual context T is replaced by a placeholder to get a modified textual context T'. Then, inspired by the Continuous Bag of Words (CBOW) Model (Mikolov, Chen, Corrado & Dean 2013), the network is trained to predict the ambiguous word w from the modified textual context T'. This unsupervised approach allows the model to learn a representation of the context of an ambiguous word. Then, in a supervised learning step, the same network is fine tuned to predict the sense label given the unmodified textual context T.

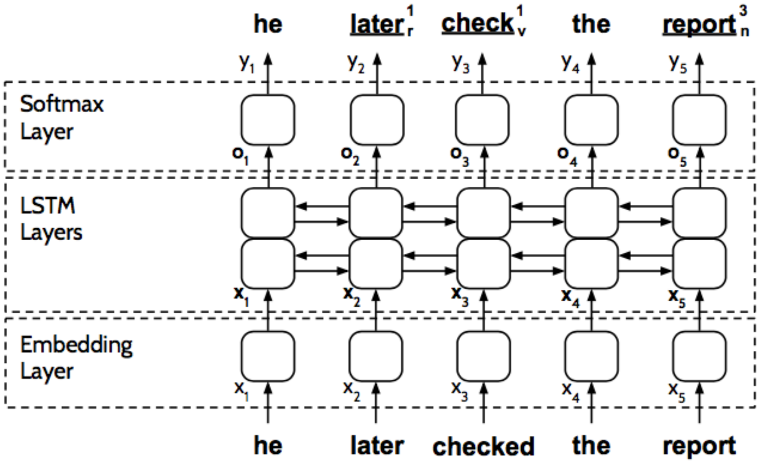

he
later^1_r
check^1_v
the
report^3_n
y_1 y_2 y_3 y_4 y_5
Softmax Layer
o_1 o_2 o_3 o_4 o_5
LSTM Layers
x_1 x_2 x_3 x_4 x_5
Embedding Layer
x_1 x_2 x_3 x_4 x_5
he
later
checked
the
report

Figure 2.3: *Bidirectional LSTM network as a 'tagger' for Word Sense Disambiguation. Each input word which is ambiguous ('later', 'checked', 'report') is tagged to its corresponding sense label. Each input word which is unambiguous ('he', 'the') is tagged to itself.*

Raganato et al. (2017) and Popov (2017) explore disambiguating all words in a sentence in a single pass. These use Bidirectional Long Short-Term Memory networks Graves et al. (2005), Graves & Schmidhuber (2005) as a 'tagger' where each word of an input sentence is being tagged to its sense as depicted in Figure 2.3. If the word is unambiguous then it is tagged to itself. If $|V|$ is the vocabulary size, and $|S|$ is the size of all senses of all words, then as a tagger the softmax is over $|V| + |S|$ classes/labels. Despite having more labels than the vocabulary, the tagger is shown to perform better than the previous state-of-the-art (Iacobacci et al. 2016). Raganato et al. (2017) also explores the use of attention mechanism over the hidden states and Recurrent Neural Network Encoder-Decoder - Neural Machine Translation architecture (Section 2.1.2, Figure 2.2) for Word Sense Disambiguation which are not found to be effective.

More recently, besides Recurrent Neural Networks, researchers are exploring transformer architecture for Word Sense Disambiguation. More specifically, contextualised representations from pre-trained Bidirectional Encoder Representations from Transformer (BERT) have been used to improve disambiguation (Hadiwinoto et al. 2019, Wiedemann et al. 2019, Scarlini et al. 2020).

2.2.3 Cross Lingual Word Sense Disambiguation

There are many different variants of the Word Sense Disambiguation task which have been explored in the Sense Evaluation (SensEval) (Edmonds 2002) and Semantic Evaluation (SemEval)[13] series of shared tasks like the classical monolingual Word Sense Disambiguation, multilingual Word Sense Disambiguation (Navigli et al. 2013), Word Sense Induction and Disambiguation (Agirre & Soroa 2007), etcetera. One important variant which has inspired our work is cross lingual Word Sense Disambiguation (Lefever & Hoste 2010, 2013).

In cross lingual Word Sense Disambiguation, the sense labels of an ambiguous word given its textual context is its translations in other languages. In the SemEval shared task, these translations are obtained from the Europarl parallel corpus (Koehn 2005) using the GIZA++ word alignment model (Och & Ney 2003). The word in the reference translation that gets aligned to the ambiguous word in the source sentence is considered as its sense label. Cross lingual Word Sense Disambiguation task was hosted when Neural Networks were not yet popular and hence neural approaches to this task have not been explored. The work presented in this thesis is among the first to explore that.

The best performing systems (Gompel & van den Bosch 2013, Rudnick et al. 2013) in cross lingual Word Sense Disambiguation shared tasks were found to be the ones using simple classification algorithms on simple Bag-of-Words[14] features/representation of the context. Gompel & van den Bosch (2013) used k-Nearest Neighbours[15] classification algorithm on bag-of-words representation of the textual context of the ambiguous word. Rudnick et al. (2013) first obtains the parallel pair in the training set which is most similar to the test sample as additional multilingual context and then used L2 Kernel Classification (Kim & Scott 2009) using bag-of-word representation of the textual context and the parallel pair. Bag-of-word representation is a very simple representation of text which ignores word order. The success of this representation over other representations shows word order is not important for resolving ambiguity in cross lingual Word Sense Disambiguation.

Another interesting approach is to use a Machine Translation system for Word Sense Disambiguation. Carpuat (2013) used a Phrase Based Statistical Machine Translation for cross lingual Word Sense Disambiguation because the sense labels are lexical translations of the

[13] `https://www.aclweb.org/anthology/venues/semeval/`

[14] Bag-of-Words is a simple representation of text. Let $V = (v_1, v_2, ..., v_n)$ be a vocabulary of unique words. For a text t, its bag-of-words representation in its simplest form is a vector $(t_1, t_2, ..., t_n)$ where $t_i = 1$ if the word v_i is found in t or else it is 0. This is a sparse representation since most of components of the vector will be zero.

[15] In k-Nearest Neighbour classification algorithm, given a vector v, we first obtain a set of k vectors from the training set which are closest to v (nearest neighbours) as measured using some distance metric. Then the class which is most common in this set of k nearest neighbouring vectors is assigned to the vector v.

ambiguous words. However, this Machine Translation system performed poorly as compared to dedicated classifiers. In the next section 2.3 we will look at more works integrating Word Sense Disambiguation and Machine Translation.

2.3 Word Sense Disambiguation in Machine Translation

Word Sense Disambiguation was originally intended to benefit Machine Translation but that has not been the case because of difficulty in integrating the two and also because of contrasting evidence that one benefits the other.

Experiments to evaluate the utility of Word Sense Disambiguation for Statistical Machine Translation was first introduced in Carpuat & Wu (2005). They used a Word Sense Disambiguation system for Chinese which predicted the correct sense of 20 Chinese words. The predicted sense was mapped to a corresponding lexical translation in English which was used by a Chinese to English word based Statistical Machine Translation system to generate an English translation of a Chinese source sentence. The disambiguation system was an ensemble of four different classifiers which used neighbouring words of the ambiguous word and their position as features. Carpuat & Wu (2005) observed a drop in Machine Translation performance after integrating Word Sense Disambiguation which suggests sense disambiguation may not be beneficial for Machine Translation. On the other hand some positive results were also seen in other efforts to integrate the two.

Vickrey et al. (2005) developed a cross lingual Word Sense Disambiguation system for French to English word-level translation task and showed positive results. Their system was a simple Logistic regression classifier[16] which used bag-of-words and position as input features. The positive result shown was for word translation and not sentence translation. Cabezas & Resnik (2005) showed marginal improvements to their Spanish to English phrase-based Statistical Machine Translation systems after integrating a cross lingual Word Sense Disambiguation system for the same language pair. Their disambiguation system used Support Vector Machines (Zhang 2010) for classification using bag-of-word input features which predicted word translations. These word translations were used for lexical selection in the decoder of their Statistical Machine Translation system. Other similar approaches have shown significant improvements to their Statistical Machine Translation systems after integrating Word Sense Disambigutaion (Carpuat & Wu 2007, Chan et al. 2007).

Both Carpuat & Wu (2007) and Chan et al. (2007) used a phrase based Word Sense Disambiguation system (or we may call it a Phrase Sense Disambiguation system) where input is an ambiguous phrase and senses are its translation (also phrases) as determined by phrase alignment information. In Chan et al. (2007), ambiguous phrase size was restricted to 2 words or less while in Carpuat & Wu (2007) the phrases could be of any size. The disambiguation system used Support Vector Machines for classification using bag-of-words, part-of-speech of each word and collocation/position of words as input features. A model (feature) estimating the probability of a translation containing the prediction of the disambiguation system was

[16] Logistic Regression classifier uses the sigmoid function σ (Appendix A.4) to estimate a probability of a class given input features. Let x be a vector of input features and W be the model weights / parameters. Then, the probability $p(c|x)$ of a class c given input features x is estimated as $\sigma(W_c \cdot x)$. Weigths / parameters are learned by maximizing $p(c_+|x)$ for the correct classes c_+ and minimizing $p(c_-|x)$ for the incorrect classes c_-.

added to the noisy channel (equation 2.6) of the phrase based Statistical Machine Translation system. Minimum Error Rate Training (Och 2003) was used to learn the weights of the model. A simpler and efficient integration was shown in Specia et al. (2008) using a n-best re-ranking technique (Och et al. 2004) where n-best translations generated by a Machine Translation system are re-ranked by adding a new feature from the Word Sense Disambiguation system to the pre-trained baseline Statistical Machine Translation model.

Unlike Statistical Machine Translation, there have been fewer attempts to integrate Word Sense Disambiguation and Neural Machine Translation. This is because, while Word Sense Disambiguation could be integrated as a sub-component of the noisy channel formulation of Statistical Machine Translation, such an integration is not possible in Neural Machine Translation. Neural Machine Translation models are expected to learn different senses of words as part of their end-to-end training of the translation task. Their Word Sense Disambiguation capabilities have been explored in Gonzales et al. (2017), Liu et al. (2018) and Marvin & Koehn (2018).

Gonzales et al. (2017) evaluate Neural Machine Translation systems on a cross lingual Word Sense Disambiguation task. They observed 70% of test samples of ambiguous words were correctly disambiguated by the translation systems. The correct sense of these samples were mostly the most frequent sense[17] of the ambiguous word. The systems mostly struggled to disambiguate samples where the correct sense of the ambiguous word was rare. Gonzales et al. (2017) proposed improving cross lingual Word Sense Disambiguation ability of a Neural Machine Translation system by passing *sense embeddings* or embeddings of *lexical chain* of the ambiguous word as an additional input feature to the encoder of the system. Sense embeddings are vector representation of senses extracted using SenseGram (Pelevina et al. 2017). Lexical chain of a word is a chain of semantically similar words within a given document which were detected using Mascarell (2017) method. Word Sense Disambiguation improvements gained were marginal and no improvements were gained in translation as measured using BLEU Papineni et al. (2002).

Marvin & Koehn (2018) proposed a methodology of exploring the Word Sense Disambiguation capabilities of the hidden representations within the encoder of a Neural Machine Translation system. To demonstrate their methodology, they considered four ambiguous words (*right, like, last* and *case*) and extracted sentences containing these words in their different senses. These sentences were then manually annotated with the correct sense of the ambiguous word and then translated into French using a English to French Neural Machine Translation system. The internal hidden activations of the different layers in the encoder of the system were extracted and the correct sense of the ambiguous word was labelled to these extracted hidden activations. The hidden activations were reduced to two dimensions using Principal Component Analysis[18] and clustered according to their sense label. These clusters were then analysed using the Dunn Index (Dunn 1973, 1974) and the Davies–Bouldin index

[17] Given an ambiguous word with multiple different senses, the most frequent sense of the ambiguous word refers to that sense which occurs most number of times in the training set.

[18] Principal Component Analysis is a transformation of a collection of points in a high-dimensional vector space to a new coordinate system such that greatest variance of the collection of points is on the first coordinate/principal component, the second greatest variance of the collection of points is on the second coordinate/principal component, etcetera. This transformation can be used for dimension reduction to k dimensions by selecting the first k coordinates/principal components which will preserve as much variance of the collection of points in the original space as possible.

(Davies & Bouldin 1979). Another way of evaluating the disambiguation capabilities of the hidden activations of the encoders was also demonstrated where they were feeded as inputs to a dedicated Word Sense Disambiguation system that used Support Vector Machines classifier. The performance of this system was then seen as a measure of the sense disambiguation capabilities of the hidden activations. The findings in Marvin & Koehn (2018), according to both the methodologies, varied because of the small sample size of sentences and ambiguous words.

Liu et al. (2018) trained an unsupervised system identical to Yuan et al. (2016) (See section 2.2.2) which maps context to an ambiguous word using a Long Short-Term Memory network. This system can be seen as learning a contextualised representation of an ambiguous word. This contextualised representation was concatenated to the embedding of every word of the source sentence which was then given as input to a Neural Machine Translation system. Liu et al. (2018) showed their Neural Machine Translation system improved at Word Sense Disambiguation and also at translation as measured using BLEU metric (Papineni et al. 2002) after adding the contextualised representations. In general, it is observed that adding more contextual information to the source sentence tends to improve the performance of a Neural Machine Translation system like adding linguistic information such as morphological features, part-of-speech tags, and syntactic dependency labels (Sennrich & Haddow 2016). One way of adding more contextual information is to look for it in other modalities like vision which we explore in the next section.

2.4 Language and Vision

Natural Language Processing tasks have traditionally been monomodal focusing only on text and textual representations while ignoring other modalities like vision and visual representations. This has changed in recent times due to (a) availability of multimodal datasets consisting both text and images (or videos) for research, and (b) advances in computer vision, more specifically in Image Classification and Object Detection, that allowed for more accurate depiction of contextual information from image (or videos) which can be used in Natural Language Processing tasks. Several multimodal Language and Vision tasks have emerged like,

- Image Captioning (Karpathy & Fei-Fei 2015, Bernardi et al. 2016), where the objective is to generate a caption (description) of an input image.

- Visual Question Answering (Antol et al. 2015), which is a multimodal extension of the text-only Question Answering task, where the objective is to give a natural language answer given an image and a natural language question about the image.

- Multimodal Machine Translation (Specia et al. 2016, Elliott et al. 2017, Barrault et al. 2018), which is a multimodal extension of the text-only Machine Translation task, where the objective is to translate a source sentence given an image as an auxiliary cue.

- Visual Sense Disambiguation (Barnard & Johnson 2005, Gella et al. 2016), which is similar to Word Sense Disambiguation except that the contextual information is from the vision modality too. In other words, given an ambiguous word and an image (and text if available), the objective is to assign the correct sense to the ambiguous word which conforms to the image.

2.4.1 From Image Classification to Image Captioning

Image Classification is the task of classifying an image into an object category depending on the presence of that object in the image. Object detection is an extension of this task where the objective is to detect multiple objects and identify their locations as determined by tight bounding boxes[19] in the image. These tasks have benefited from the ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al. 2015) which consists of more than 1.2 Million images in which objects have been annotated with 1000 object classes using crowd-sourcing to collect annotations at a large scale (Deng et al. 2009, Su et al. 2012, Russakovsky et al. 2013, Deng et al. 2014).

Previously, a two step approach was adopted to solve the Image Classification problem. First, handcrafted features were extracted from images, and then these were given as input to a trainable classifier. In this two step approach, the accuracy of the classification task was largely dependent on the design of the feature extraction method which invariably proved to be very difficult. This changed with the rise of Deep Convolutional Neural Networks which is a one step process in which feature extraction is also learnt by the network. Krizhevsky et al. (2012), Hinton et al. (2012) developed an Image Classification system using Deep Convolutional Neural Networks and won both the Image Classification and the Object Detection tasks in the 2012 ImageNet Large Scale Visual Recognition Challenge. Since then, most of the submissions to the ImageNet challenge, all its winners and popular state-of-the-art systems are all based on Deep Convolutional Neural Networks (Zeiler & Fergus 2013, 2014, Simonyan & Zisserman 2015, Szegedy et al. 2015, He et al. 2016, Girshick et al. 2014). He et al. (2015) was the first to outperform humans at the Image Classification task as set out in the ImageNet challenge and several more subsequent systems have surpassed human-level performance. Hence, to the extent of the ImageNet challenge, the Image Classification and Object detection tasks are considered to be solved.

Central to Convolutional Neural Networks is the concept of *convolution filters* which extract features from the image. We now provide a formal definition of a convolution filter: An image, in its simplest form ignoring colours, is a matrix of pixels. Let I be an image, then $I(x, y)$ which is $(x, y)^{th}$ element of the of the Image matrix represents the brightness of the pixel located at coordinates (x, y). Let I be an $m \times n$ matrix. A convolution filter K, also known as 'kernel', is a matrix of weights or parameters. It is a smaller matrix than the image, usually a 3×3 matrix in practice which is what we have assumed in our definition too. We then define a simple *convolution product* $\times_c$ between the Image and the convolution filter that results in a new $(m-2) \times (n-2)$ matrix, which we will denote as C, as follows:

$$I \times_c K = C$$

where

$$C(x, y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} I((x+1)+i, (y+1)+j) \cdot K(2+i, 2+j) \tag{2.15}$$

[19] An imaginary rectangle that engulfs the object within it.

Depending on the weights of the convolution filter, it is possible to detect different features, like say a vertical edge or a horizontal edge, in a 3×3 portion of the image. Convolution product can be thought of transforming the $m \times n$ image I to a $(m-2) \times (n-2)$ convolution matrix C which consists only that feature which the convolution filter K detects in the image. For example, if the convolution filter K has weights which enables it to detect horizontal edges in the image I, then the convolution matrix C $(I \times_c K)$ will consist of all the horizontal edges in the image as depicted in Figure 2.4.

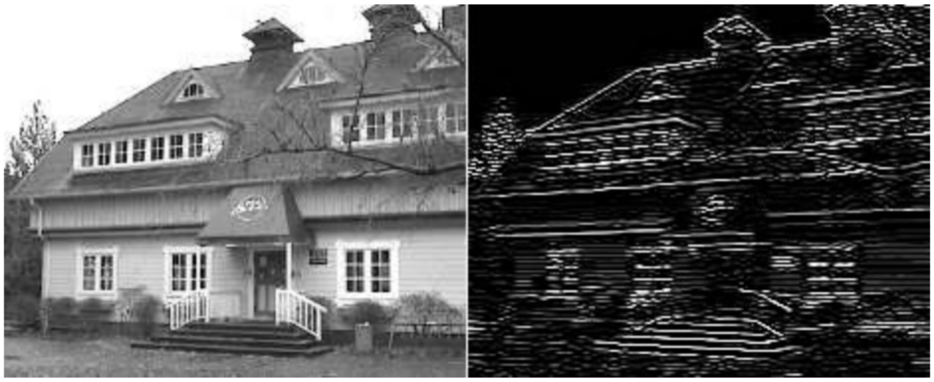


Figure 2.4: *A Convolution Filter which detects horizontal edges transforms an image (left) to a Convolution Matrix or Feature Map (right) consisting the horizontal edges of the image.*

The weights of the convolution filters are learnt when training the Convolutional Neural Network using the backpropagation algorithm. Usually, many different convolution filters $(K_1, K_2, ..., K_d)$ exist to extract different features from the image generating many different convolution matrices $(C_1, C_2, ..., C_d)$. Usually, a non-linear activation function (Appendix A.4) is then applied to the convolution matrices to get what are known as feature maps which we are also denoting as C_s. The feature maps C_s can be regarded as another image and more convolution filters can be then applied on these forming subsequent layers of convolutions. Proceeding this way, in 'Deep' Convolutional Neural Network, we have many layers of convolutions stacked one over the other. In an extreme case, ResNet-1202 (He et al. 2016) has 1201 layers of convolutions stacked one above each other.

In a fully trained Deep Convolutional Neural Network, the first layers of convolutions detect low level features like edges. The next layers of convolutions detect higher level features like shapes (circle, rectangle, etc). Going deeper, the next layers of convolutions detect parts of an object like eyes or wheels. Finally, the deepest layers of convolutions are able to detect faces and other objects in the image like cars, elephants, etcetera (see Figure 2.5). In addition to convolution filters, state-of-the-art Deep Convolutional Neural Networks also consist other important engineering aspects like Pooling (Zeiler & Fergus 2013), Residual connections (He et al. 2016), etcetera.

In my personal opinion, I perceive the different layers of a Deep Convolutional Neural Network in terms of capturing contextual image information at different 'levels of abstraction', from pixels to edges to geometric shapes to parts of an object to full objects and their location in the image, in the form of feature maps. According to me, lower levels of abstraction

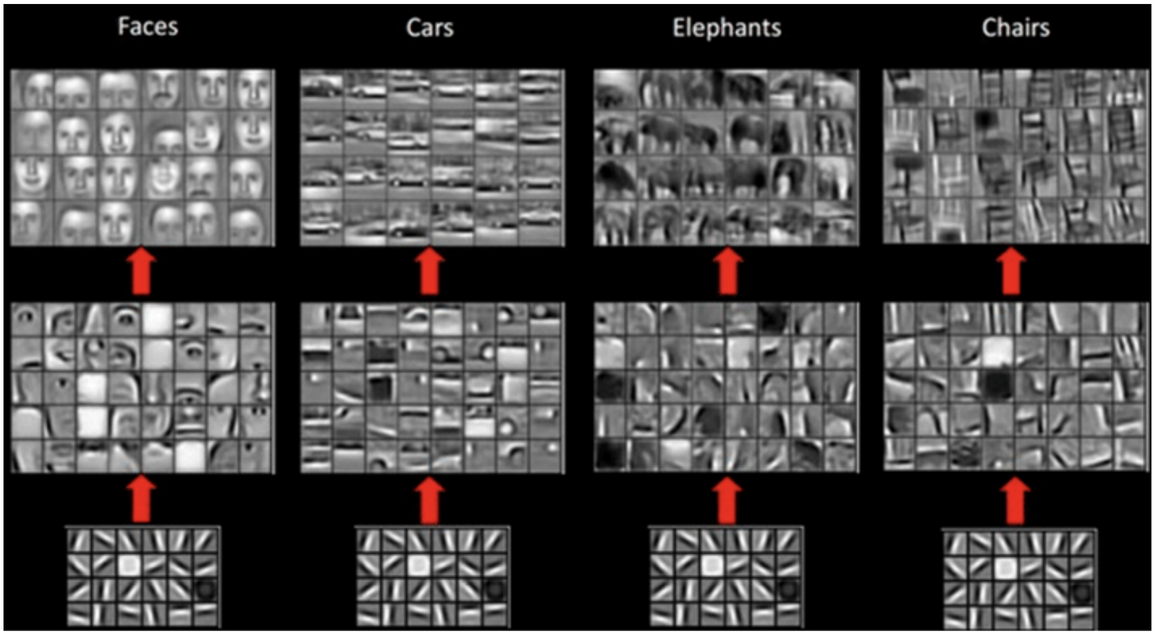


Figure 2.5: *Different layers of convolution in a Deep Convolutional Neural Network detect different features. First layers of convolution detect low level features like edges. Deeper layers of convolutions detect certain shapes like an eye or a wheel. Deepest layers of convolution detect high level features like faces or cars.*

like pixels or edges or even geometric shapes to some extent are not particularly useful for Natural Language Processing because Human languages tend to operate at higher levels of abstraction, usually starting from parts of objects (eyes) to full objects (face), which are nouns, and beyond like action (smiling) which are verbs and characteristics of the object ('beautiful' face) which are adjectives, so on. In recent times the feature maps from the deeper layers of Image Classification models are capturing contextual information from the images at the levels of abstraction which is overlapping with the levels of abstraction where human languages operate, we are beginning to see the two disjointed fields of research 'Computer Vision' and 'Natural Language Processing' as merging (see Figure 2.6). I also believe that for a more meaningful and useful integration of the two fields, we need models which can capture contextual information from the images at even higher levels of abstraction venturing further into the range where human languages actually operate.

The first effective merger between Computer Vision and Natural Language Processing was demonstrated in Image Captioning in Vinyals et al. (2015), Xu et al. (2015). Their systems extracted features from an image I in the form of feature maps from the deeper convolution layers, usually the penultimate layer, of a Deep Convolutional Neural Network trained on the ImageNet challenge. These extracted features C, which are feature maps representing contextual information from the image at higher levels of abstraction, were then fed to a Recurrent Neural Network decoder which generated a caption (description) of the image. Formally, let I be the raw image in its pixel form, C be the contextual information from the image like feature maps capturing objects and their location, and S be a sentence describing the image, then the Convolution Neural Network encoder can be viewed as modelling the

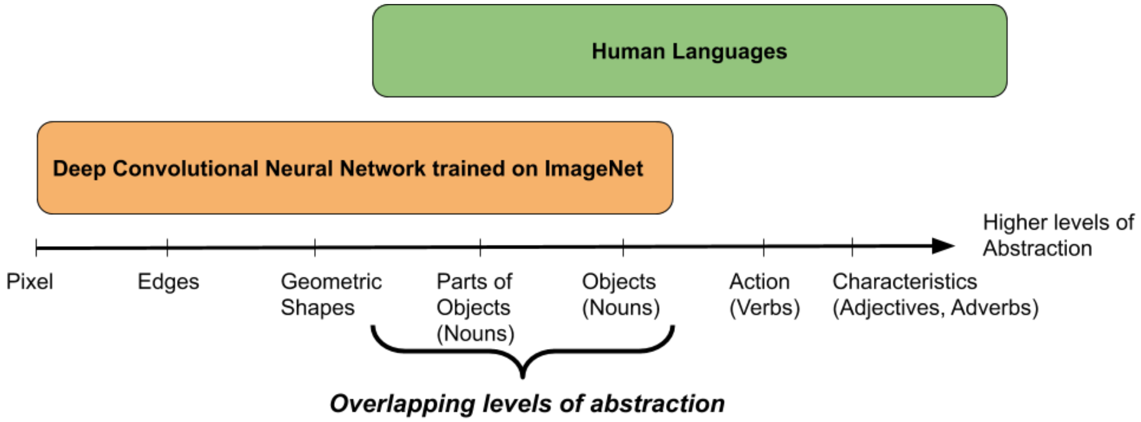

Human Languages
Deep Convolutional Neural Network trained on ImageNet
Higher levels of Abstraction
Pixel
Edges
Geometric Shapes
Parts of Objects (Nouns)
Objects (Nouns)
Action (Verbs)
Characteristics (Adjectives, Adverbs)
Overlapping levels of abstraction

Figure 2.6: *Spectrum of Abstraction with different Levels. Deep Convolutional Neural Networks trained on ImageNet dataset capture contextual information from the image at different levels of abstraction from pixel-level to object-level. Human languages tend to operate from level of objects or parts of object onwards to higher levels of abstraction. The overlapping levels of abstraction has allowed for the two disjointed fields of research - Computer Vision and Natural Language Processing - to merge in recent times.*

context given the image as $p(C|I)$ and the Recurrent Neural Network decoder can be viewed as modelling the description given the context $p(S|C)$.

Vinyals et al. (2015) used GoogLeNet or Inception v1 (Szegedy et al. 2015) and Xu et al. (2015) used the Oxford VGGnet (Simonyan & Zisserman 2015) as image encoders which were trained on the ImageNet challenge. Their decoder is a single Long Short-Term Memory unit. Xu et al. (2015), in addition, used attention mechanism as described in equations 2.9,2.10 and 2.11 (see Section 2.1.2). The attention is over image representation from convolution layers in VGGnet. These models were trained and tested on the Pascal VOC (Farhadi et al. 2010), Flickr8k (Rashtchian et al. 2010), Flickr30k (Young et al. 2014), MSCOCO (Lin et al. 2014) and SBU (Ordonez et al. 2011) datasets consisting of pairs (I, S) of images and its descriptions. These systems defeated previous state-of-the-art (SOTA) Image Captioning systems and achieved near human like performance of generating captions (see Table 2.1) as measured using BiLingual Evaluation Understudy (BLEU) which compared the generated description with a reference description. The descriptions generated by these systems were also found to be fluent. Further analysis reveals 27% of the descriptions generated by Vinyals et al. (2015) are better than or equal to humans, and 32% of the generated descriptions pass the Turing Test[20] (Vinyals et al. 2016). The success of Vinyals et al. (2015), Xu et al. (2015) and other Image Captioning systems can be attributed as one of the reasons leading to the rising interest of the research community in joint Language and Vision tasks. For a more detailed description of different Image Captioning approaches, please refer Bernardi et al. (2016).

It is important to note at this stage that one major problem faced in many Language and Vision tasks is the evaluation of multimodal systems. For instance, in Image Captioning,

[20] In other words, humans were unable to decide if these captions (descriptions) were generated by a computer or a human.

Image Captioning Approach	PASCAL	Flickr30k	Flickr8k	SBU	MSCOCO
Previous SOTA	25	56	58	19	-
Vinyals et al. (2015)	59	66	63	28	67
Xu et al. (2015)	-	67	67	-	72
Human	69	68	70	-	-

Table 2.1: *Performance of Image Captioning approaches as measured using BLEU-1 across different Image Captioning test sets.*

systems were evaluated using Machine Translation metrics like BLEU, METEOR, TER or other Image Captioning specific metrics like CIDEr (Vedantam et al. 2015), SPICE (Anderson et al. 2016) or ROUGE (Lin 2004). These may not be completely relevant because such metrics disregard the actual image information. One solution is to develop and use new image-aware evaluation metrics for Language and Vision tasks like VIFIDEL (Madhyastha et al. 2019) for Image Captioning which measures similarity between the objects in the image and the words in the generated descriptions using Word Mover's Distance (Kusner et al. 2015).

Secondly, another important research question is to probe the utility of images and image representations for Language and Vision tasks. A standard practice in many Language and Vision tasks, like in Image Captioning, is to extract features from the state-of-the-art Deep Convolutional Neural Networks like Simonyan & Zisserman (2015), Szegedy et al. (2015) and He et al. (2016) which are trained for the ImageNet's Image Classification challenge where the output of the network is a distribution over 1000 object categories. The features commonly used can be of three types (a) Spatial features which are the feature maps extracted from specific convolutional layers, (b) Pooled features which is what we get when a pooling layer or a fully connected layer is applied to the feature maps which vectorizes the matrices and in the process looses spatial information, and (c) Posterior distribution over object classes which is extracted from the output layer (See figure 2.7). These can be viewed as an ascending level of abstractions from dense representation to compact vector representation to objects in the image. In addition to these, we can also obtain pooled feature vectors from different regions of a given image, with regions predicted by an Object Detection systems like Girshick et al. (2014), Ren et al. (2015).

In Image Captioning, Madhyastha et al. (2018) explored the usefulness of different image representations like (1) Pooled image features from the penultimate layer of a Deep Convolution Neural Network trained on ImageNet challenge, (2) Posterior distribution over object classes from the final layer of a Deep Convolutional Neural Network, (3) Bag-of-Objects which are the different objects found in the image. They found Bag-of-Objects representation of the image, which is a higher-level of abstraction, benefits Image Captioning. Wu, Shen, Liu, Dick & Van Den Hengel (2016) and You et al. (2016) also found a similar result where they show a posterior distribution over object classes benefits Image Captioning more than using feature maps from lower layers of a Deep Convolutional Neural Network. Wang et al. (2018) show that more high-level abstractions like the frequency, size and position of objects in the image can also play a role in forming a good image representation which is useful for Image Captioning. In general, these results suggest contextual information from images capturing

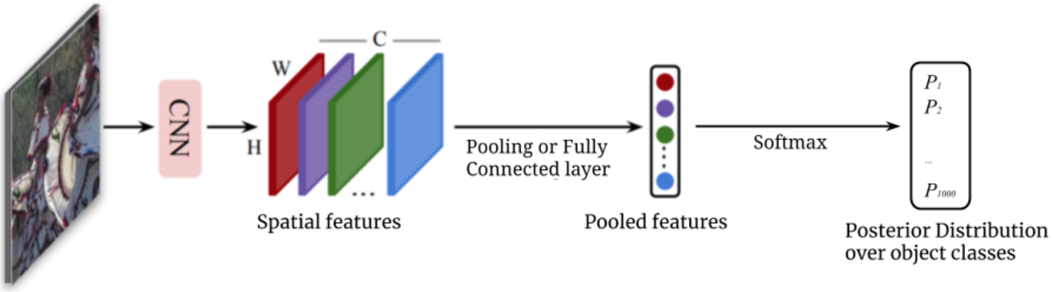

CNN
C
W
H
Spatial features
Pooling or Fully Connected layer
Pooled features
Softmax
P_1
P_2
...
P_{1000}
Posterior Distribution over object classes

Figure 2.7: *Image Features from a Deep Convolutional Neural Network trained on Image Classification. Spatial features preserve spatial information while pooled features and posterior distribution over object classes lose spatial information.*

higher levels of abstraction are useful, which is something we also found in our experiments on Multimodal Word Sense Translation.

2.4.2 Multimodal Machine Translation

The success of Image Captioning encouraged researchers to extend monomodal text-only Natural Language Processing tasks to incorporate Vision modality. One such extension is Multimodal Machine Translation. Formally, if I is an image, x is its description in the source language and Y is all possible translations in the target language, then a probability distribution $p(y|x, I)$ could be used to get the correct translation $\hat{y}$ as follows:

$$\hat{y} = \arg\max_{y \in Y} p(y|x, I) \tag{2.16}$$

To enable modelling $p(y|x, I)$, we need a corpus of triples of the form (x, y, I). There have been a few such datasets like IAPR TC-12 (Grubinger et al. 2006, Clough et al. 2006), extension of Flickr8k to Chinese (Li et al. 2016), extension of Flickr8k to Turkish (Unal et al. 2016), and Multi30k dataset[21] (Elliott et al. 2016, Specia et al. 2016, Elliott et al. 2017, Barrault et al. 2018). This PhD research is based on Multi30K which was created from the Flickr30k dataset (Young et al. 2014).

Multi30K dataset: The Multi30K dataset originally contained 31,014 images described in English with translations in German (Elliott et al. 2016). The images were sampled from Flickr30K dataset (Plummer et al. 2015). The English descriptions were collected from Amazon Mechanical Turk[22] and the German translations were collected from professional English-German translators contracted via an established language service company in Germany. To ensure an even distribution over description length, the English descriptions were chosen based on their relative length, with an equal number of longest, shortest, and median length source descriptions. The translators were shown an English language sentence and asked to produce a correct and fluent translation for it in German, without seeing the image which can be regarded as a caveat of this dataset. The original creators of the dataset had

[21] `https://github.com/multi30k/dataset`

[22] `http://www.mturk.com`

decided against showing the images to translators to make this process as close as possible to a standard translation task. In Specia et al. (2016), this dataset was split into 29,000 Train samples, 1014 Validation samples and 1000 Test samples (which we will refer to Test 2016).

En: *a group of men are loading cotton onto a truck*

De: *eine gruppe von männern lädt baumwolle auf einen lastwagen*

Fr: *un groupe d'hommes chargent du coton dans un camion*

Cz: *skupina mužů nakládá bavlnu na náklaďák*

Figure 2.8: *A Multi30K sample consisting an image and its description in English with translations in German, French and Czech.*

Dataset	Statistic	English	German	French	Czech
Train (29000 samples)	Number of words	377534	360706	409845	297212
	Words per sentence	13.0	12.4	14.1	10.2
Validation (1014 samples)	Number of words	13308	12828	14381	10342
	Words per sentence	13.1	12.7	14.2	10.2
Test 2016 (1000 samples)	Number of words	12968	12103	13988	10497
	Words per sentence	13.0	12.1	14.0	10.5
Test 2017 Flickr (1000 samples)	Number of words	11376	10758	12596	9078
	Words per sentence	11.4	10.8	12.6	9.1
Test 2017 MSCOCO (461 samples)	Number of words	5239	5158	5710	4115
	Words per sentence	11.4	11.2	12.4	8.9
Test 2018 (1071 samples)	Number of words	13774	12325	15564	10684
	Words per sentence	12.8	11.5	14.5	10.0

Table 2.2: *Statistics of the Multi30K dataset which consists 5-tuples consisting an image, its description in English, and its translations in German, French and Czech.*

Later, in Elliott et al. (2017), the dataset was extended to include (1) crowdsourced French translations, and (2) two additional test sets - Test 2017 Flickr and Test 2017 MSCOCO. The crowdsourced translations were collected from 12 workers using an internal platform where the translators had access to the source English description, the image and an automatic translation created with a standard phrase-based Statistical Machine Translation system (Koehn et al. 2007) trained on WMT15 parallel text (Bojar et al. 2015). The automatic translations were presented to the crowdworkers to further simplify the crowdsourcing task. A caveat for these translations is that the crowdworkers were not professional translators.

The Test 2017 Flickr has 1000 images sampled from the Flickr dataset with crowdsourced

English descriptions using crowdflower[23] and professional German translations via professional English-German translators and crowdsourced French translations by the same 12 workers who extended the Train, Validation and Test 2016 versions of Multi30K. The Test 2017 MSCOCO has 461 images sampled from the VerSe dataset (Gella et al. 2016) which comes from the MSCOCO dataset (Lin et al. 2014) and TUHOI dataset (Le et al. 2014). These come with English descriptions which consist ambigous verbs (action words with multiple senses). Its German and French translations were created using the same procedure as Test 2017 Flickr dataset.

Finally, in Barrault et al. (2018), the Multi30K dataset was extended further with (1) translations of the image descriptions into Czech and (2) A new test dataset - Test 2018. The Czech translations were produced by 15 workers who were university and high school students and teachers. The translators used the same internal platform that was used to collect the French translations for the Multi30K dataset. The Czech translators had access to the source description in English and the image only (no automatic translation into Czech was presented). The Test 2018 contains images sampled from Flickr dataset and then crowdsourced English description and professional German translation and crowdsourced French and Czhech translations consistent to the earlier versions of the dataset.

Today, the Multi30k dataset consists of 5-tuples of the form (en, de, fr, cs, I) where I is an image, en is a description of the image and de, fr, cs are the translations of the description in German, French and Czech respectively (see Figure 2.8). The dataset is divided into training, validation and different test sets (see Table 2.2) for the different years the Multimodal Machine Translation shared task was conducted.

A standard practice in Multimodal Machine Translation, inspired from Image Captioning, is to extract image features of an image using a Deep Convolutional Neural Network trained on ImageNet challenge (spatial features or pooled features or posterior distribution over object classes) and then use it in a standard Recurrent Neural Network based Encoder-Decoder. Some of the approaches of using the image features are described below:

Use Image Features to initialise the Encoder and Decoder: This is a very easy and straightforward incorporation of image features in the Neural Machine Translation architecture employed in several works (Elliott et al. 2015, Huang et al. 2016, Hokamp & Calixto 2016, Libovický et al. 2016, Calixto, Dutta Chowdhury & Liu 2017, Madhyastha et al. 2017, Caglayan, Aransa, Bardet, Garcia-Martinez, Bougares, Barrault, Masana, Herranz & van de Weijer 2017). The only care that needs to be taken is that the hidden state dimension of the encoder or decoder and the dimension of image features is equal. Formally, if $\mathbf{v}$ is the image feature of dimension d_v, and the Recurrent Neural Network Encoder or Decoder takes a hidden state $\mathbf{h}$ of dimension d, then some transformation (linear or non-linear) $f : \mathbf{R}^{d_v} \rightarrow \mathbf{R}^d$ is learned which is then followed by initializing the start state $\mathbf{h}_0$ of Recurrent Neural Network simply as:

$$\text{Hidden State of Encoder or Decoder at time step } 0 = \mathbf{h}_0 = f(\mathbf{v}). \tag{2.17}$$

The rest of the Neural Machine Translation architecture remains the same. More often than not, dimension of image features d_v is much larger than the dimension d of hidden state

[23] http://www.crowdflower.com

of Encoder or Decoder . This means the transformation f is shrinking the image features further which is essentially dimension reduction. A potential weakness could therefore be that some valuable information from the image is being lost in this dimension reduction step.

Add or Multiply of Concatenate the Image Feature to Word Embeddings: This is another simple strategy of using image features in Neural Machine Translation. Madhyastha et al. (2017) added image features to word embeddings of the words in the source sentence. Caglayan, Aransa, Bardet, Garcia-Martinez, Bougares, Barrault, Masana, Herranz & van de Weijer (2017) did element-wise multiplication of image features to word embedding of words in the source sentence and target sentence. There is a possibility of concatenating too. Basically, if w is a word-embedding being used in the Neural Machine Translation system (at any stage), then we can combine w and $\mathbf{v}$ in some way like adding, multiplying and concatenating. A generalization would be to have some transformations f_1 and f_2 such that $f_1(w)$ and $f_2(\mathbf{v})$ are compatible for a combination like addition or multiplication. Again, dimensions of the word embeddings and the visual features need to be equal (except in concatenation case) or at least compatible to the operation to be tried. In practice, there is always a dimension reduction of the visual features which takes us back to the loss of information mentioned earlier. A big advantage though is that some form of multimodal embedding for each word given image is being learned like in addition

$$f_1(w) + f_2(\mathbf{v}) = f(w, \mathbf{v}) = \text{Multimodal Embedding} \tag{2.18}$$

This, in theory, has the potential of disambiguating the senses of ambiguous words w using image $\mathbf{v}$.

Use Image Feature as a word in the source sentence: This strategy treats image feature as a word. Calixto, Dutta Chowdhury & Liu (2017) adds image features as words at the beginning and end of the source sentence. Basically, if the sentence is a sequence of word embeddings $(w_1, w_2, \cdots, w_n)$, then we feed the following sequence of embeddings to the Neural Machine Translation system as inputs:

$$\text{Input to Recurrent Neural Network Encoder} = (f(\mathbf{v}), w_1, w_2, \cdots, w_n, f(\mathbf{v})) \tag{2.19}$$

This is, theoretically, no different from initializing the Encoder recurrent neural network, except that it is also ending with the image features. In general, we can place the image feature at any position in the source sentence. This approach also requires image features to be of the same dimension as the word embeddings which usually involved dimension reduction of the image, like in previous approaches. So, this approach is also likely to have loss of visual information. One interesting possibility could be to look at strategically placing the visual features in the sentence. By strategically, it means to place the image features only at places where we may need them like around ambiguous words only.

Multitask Learning - Use Image Features in a separate task: Elliott & Kádár (2017) introduced the idea of multitask learning for Multimodal Machine Translation. In this approach we have a single Neural Network performing two tasks, one of which is the standard translation task and the other uses the image feature. The network is trained end-to-end. In Elliott & Kádár (2017), the other task is the reverse of Image Captioning which is to

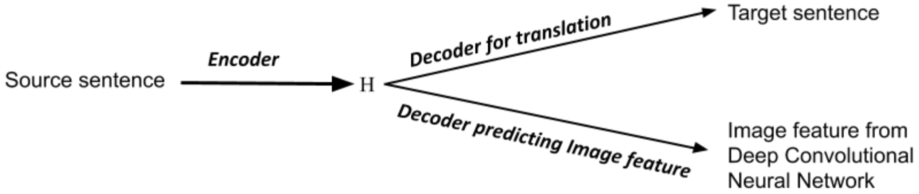

Source sentence
Encoder
H
Decoder for translation
Target sentence
Decoder predicting Image feature
Image feature from Deep Convolutional Neural Network

Figure 2.9: *Multi-task Learning (Imagination) for Multimodal Machine Translation. One Encoder encodes the source sentence to H and two decoders, one decodes H into the translation and ohter decodes H into the image feature which has been derived from a Deep Convolutional Neural Network trained on Image Classification.*

predict image features from the source sentence description as depicted in Figure 2.9. This architecture has one Recurrent Neural Network Encoder which encodes the source sentence and two Decoders, a Recurrent Neural Network Decoder for translation and a Decoder with just a softmax layer for predicting the image feature. It is an interesting strategy with the objective of using visual features in regularizing the Encoder weights. Unlike other Multimodal Machine Translation approaches where the number of model parameters increase due to the inclusion of image features, here the number of parameters that go into the standard Machine Translation task remain the same. The additional parameters due to image features are only for the other task. The big disadvantage of this approach, however, is that images have no role to play at test time. Consider the hypothetical situation where the source sentence is *"a man is holding a seal"* (The same example in the introduction. See Figures 1.2 and 1.3). Since image will not be used as an input by such a model at test time, it will therefore be, theoretically, incapable of resolving the ambiguity of 'seal'.

Separate Attention Mechanism and Gating over Image Features: The idea of separate attention mechanism over image features for Multimodal Machine Translation was first introduced in Calixto, Liu & Campbell (2017). In a text-only Neural Machine Translation, attention mechanism is over the hidden states of the Encoder (see equations 2.9, 2.10 and 2.11). Here the attention mechanism is over the spatial features of the image. The two attention mechanisms, one over Encoder hidden states and one over visual spatial features, result in two context vectors - textual context $\mathbf{c}_t^T$ and visual context $\mathbf{c}_t^V$ - at each time step t for the Decoder Recurrent Neural Network. The two context vectors can be combined in many different ways like concatenating or adding. One interesting approach is to have a second attention mechanism over the two context vectors (Helcl & Libovický 2017), to get one combined context vector. Such a model architecture allows the Decoder to peek at different areas in the image depending on what words it has generated so far. A big positive of this approach is that it seems to be doing what a human translator does, i.e. when generating the next word in the translation look at the image and the source sentence to verify. A potential future idea could be to have three attention mechanisms - (a) Attention over source text, (a) Attention over image, and (a) Attention over translation/target sentence generated so far. The engineering practicality of this idea needs to be looked into, but the analogy is that a human translator would look at all three - the source sentence, the image, the translation made so far - to generate the next word. An important feature of attention mechanisms worth mentioning is the concept of 'Gating', i.e. a sigmoid function that decides to allow the signal to pass or not. The analogy is that a human translator has the freedom to disregard the image when not needed in generating the next word. More discussion on the importance of gating in such attention based multimodal machine translation approaches can be found in Delbrouck & Dupont (2017).

We will now take a closer look at how Multimodal Machine Translation systems are evaluated. Currently, Multimodal Machine Translation systems are evaluated using the same automatic metrics which are used in evaluating text-only machine translation systems. *The closer a machine translation output is to a professional human translation, the better it is.* This is the central idea underlying most metrics. The popular ones are BLEU (Papineni et al. 2002), METEOR (Denkowski & Lavie 2014) and TER (Snover et al. 2006).

BLEU: To measure the 'closeness', BLEU relies on computing n-gram precision p_n which is essentially the proportion of n-grams in machine translation output found in the reference (without repetition). Consider the following example:

Candidate: *The cat is on the mat*

Reference: *There is a cat on the mat*

Here Unigram precision is $p_1 = 5/6$, bigram precision is $p_2 = 2/5$, and so on. The final formulation of BLEU metric is some form of parametrized geometric mean of different n-gram precisions as follows:

$$BLEU = BP \times exp(\sum_{n=1}^{N}(w_n log(p_n))) \tag{2.20}$$

Where BP is the penalty term to penalize the score of shorter candidate translations, N is the number of n-gram precisions to be considered and w_n are weights that sum to 1. If c is length of candidate sentence and r is the length of reference, then typically,

$$N = 4 \qquad w_n = 1/N = 1/4 \qquad BP = \begin{cases} 1, & \text{if } c \leq r \\ e^{1-(r/c)}, & \text{if } c \geq r \end{cases} \tag{2.21}$$

The advantage of the above formulation is that it is easy, quick and inexpensive to compute. The disadvantage, however, is that BLEU metric completely disregards use of synonyms and only measures direct word-by-word similarity, looking to match and measure the extent to which word clusters in candidate and reference are identical. Accurate translations that use different words score poorly since there is no match in the reference. Callison-Burch et al. (2006) showed improvements in BLEU do not necessarily indicate achieving actual improvements in translation quality. Over the years, many have argued and criticized using BLEU to evaluate machine translation systems and thus new metrics have emerged like Meteor.

METEOR: METEOR evaluates candidate translations by aligning them to reference translations and calculating sentence-level similarity scores. For a candidate-reference pair, the space of possible alignments is constructed by exhaustively identifying all possible matches between the sentences according to the following matchers - (1) **Exact:** Match words if they are identical. (2) **Stem:** Stem words using a language appropriate Snowball Stemmer (Porter 2001, Willett 2006) and match if the stems are identical. (3) **Synonym:** Match words if they are found to be in the same synonym set according to WordNet database (Fellbaum 1998) or other language specific databases. (4) **Paraphrase:** Match phrases if they are listed as paraphrases in a language appropriate paraphrase table. Paraphrases are automatically extracted from the training parallel sentence using the translation pivot approach (Bannard & Callison-Burch 2005). After aligning candidate sentence to reference, the METEOR score is calculated as follows. First, content and function words are identified in the candidate/hypothesis denoted as h_c, h_f respectively and also in the reference denoted as r_c, r_f respectively. For a list of matchers being used (in this case four matchers are used), let m_i denote the i^{th} matcher, and $m_i(h_c)$ be the counts of the number of content words covered by matches of this type in the hypothesis. Similarly $m_i(h_f), m_i(r_c)$, and $m_i(r_f)$ are counted. Following this a weighted precision P and recall R are computed as follows:

$$P = \frac{\sum_i w_i(\delta m_i(h_c) + (1-\delta)m_i(h_f))}{\delta|h_c| + (1-\delta)|h_f|} \tag{2.22}$$

$$R = \frac{\sum_i w_i(\delta m_i(r_c) + (1-\delta)m_i(r_f))}{\delta|r_c| + (1-\delta)|r_f|} \tag{2.23}$$

where w_i is the weightage of the i^{th} matcher and δ is the weightage given to content words. Finally, Meteor is calculated as a parametrized harmonic mean of precision and recall together with a penalty term *pen* as follows,

$$METEOR = \frac{(1-pen) \times P \times R}{\alpha P + (1-\alpha)R} \tag{2.24}$$

The penalty term accounts for gaps and differences in word order and it is described in detail in Denkowski & Lavie (2014). The advantage of METEOR is that it directly addresses many weaknesses of BLEU like (a) lack of recall, (b) disregarding synonyms, (c) geometric mean of n-gram precisions can be 0 and hence BLEU is meaningless at sentence level. The disadvantage, however, is that computing METEOR is complicated with many hyperparameters ($\alpha, \delta,$ matcher weights $w_i,$ etc.) and that it uses external resources like WordNet for synonym. It cannot be used for languages that do not have WordNet like databases like Persian for instance. Also, the overall computation is slower than BLEU.

TER: Translation Edit Rate (TER) has a post-editing approach to evaluate Machine Translation. It measures the amount of editing that a human would have to perform to change a system output so that it matches a reference translation. Its formulation is rather simple:

$$\text{TER} = \frac{\text{Number of edits}}{\text{Length of reference translation}} \tag{2.25}$$

The kind of edits include *insertion*, *deletion*, and *substitution* of single words as well as shifts of word sequences. Punctuation changes and capitalization are also considered as edits.

BLEU, METEOR and TER rely on gold standard reference translation and could suffer from 'reference bias' (Fomicheva & Specia 2016) as these metrics only look at the monolingual reference translation and not the input sentence or the image. Besides, my personal observation is that the influence of image on the translation is generally subtle but nevertheless quite valuable. In the sense, if a human translator was asked to translate a source sentence first and only then is allowed to look at the corresponding image and make changes, then the changes or edits that will be made are too few, albeit quite valuable because the meaning of the whole sentence changes radically. Often no change is needed, but whenever it is needed then changing one or two words is usually more than enough. A proper survey study is needed to provide evidence for this observation, but in a toy experiment with just 50 examples we saw as low as 0.4% of words changed (a generous upper bound). This amounts to atmost 0.4% improvement in unigram precision, 0.8% improvement in bi-gram precision, 1.2% in 3-gram and 1.6% in 4-gram. Now from Mathematics, we know geometric mean is smaller than arithmetic mean which essentially means, improvements in BLEU-4 that an image can contribute will have an upperbound of 1% (a generous upperbound). In practice, the improvements we observe will be much lower and often not statistically significant. This essentially highlights the limitation of automatic machine translation metrics being incapable to capture the improvements in translation due to images. Since the impact of images on Machine Translation is subtle according to our toy experiment, hence more sensitive metrics are needed. Metrics which, perhaps, focused on specific improvements an image can bring to translation like disambiguating ambiguous word correctly and then translating it accordingly.

2.4.3 Visual Sense Disambiguation

Visual Sense Disambiguation is the task of disambiguating the sense of a word from an image. Formally, given an image I, an ambiguous word w, its textual context T and a set of its senses $S(w) = \{s_1^w, s_2^w, ..., s_{|S|}^w\}$, the objective is to learn a mapping f such that $f(w, I) \in S(w)$ is the most appropriate sense of the word w. We can use a probability distribution $p(s|w, T, I)$ to identify the correct sense just like in equation 2.13 as follows:

$$f(w, I, T) = \hat{s} = \arg\max_{s \in S(w)} p(s|w, I, T) \tag{2.26}$$

Modelling $p(s|w, I, T)$ remains an important challenge because there have been very few attempts and fewer datasets for solving this task.

The first approach to Visual Sense Disambiguation was done for nouns using web images in Loeff et al. (2006). The images were extracted from Yahoo! image query API using the ambiguous words as keywords to query. The extracted images were then labeled with the correct sense by human annotators via crowdsourcing. It is important to note, Alm et al. (2006) observed that annotating sense labels to images is more difficult, subjective and vague as compared to annotating sense labels to text. Next, multimodal features from the image and text on the webpage containing the image were extracted. For text features, bags-of-words weighted with TFIDF (Appendix A.9) was used. For image features, they identified key regions in the image using keypoint detection algorithm Kadir & Brady (2001) and then identified the keypoint to be a class from a collection of 300 classes using a Gaussian Mixture model (Reynolds 2009). Then bag-of-keypoints was used as the image feature. Next, they used spectral clustering (Ng et al. 2002) over the multimodal features to cluster the images and identify the senses. The most common sense in a cluster was identified as the sense of that cluster.

In other approaches, Saenko & Darrell (2009) employ a Lesk-based approach (see equation 2.12) where sense definition $D(s)$ is obtained from a dictionary and words surrounding the image on the webpage containing it is used as the context T. One may argue that this does not count as Visual Sense Disambiguation. Barnard & Johnson (2005) and Chen et al. (2015) use Object Detection models to extract the image features. Barnard & Johnson (2005) then uses a statistical noisy-channel formulation like in equation 2.14 and the Expectation Maximization algorithm (Dempster et al. 1977). Chen et al. (2015), in addition to image features, extracts textual features of the text surrounding the image from the webpage containing it. Both the text features and image features are taken as pairs and then co-clustered using Expectation-Maximisation algorithm (Dempster et al. 1977).

Finally, the most recent work in Visual Sense Disambiguation (Gella et al. 2016) focused on Visual Sense Disambiguation of verbs and created a dedicated dataset VerSe[24] for the task. Their approach is also a Lesk-based algorithm (see equation 2.12) which measures relatedness between image features or multimodal features and text features of the dictionary definition of senses. The image features are obtained from VGGnet (Simonyan & Zisserman 2015). All the approaches in Visual Sense Disambiguation, mentioned above, have been unsupervised approaches; either a Lesk-based approach or Clustering. Our approaches to Multimodal Word Sense Translation, which can be regarded as a version of Visual Sense Disambiguation, are supervised approaches.

[24] `https://github.com/spandanagella/verse`

Chapter 3

Dataset for Multimodal Word Sense Translation

Inspired by the tasks of Multimodal Machine Translation and Visual Sense Disambiguation we introduced a new task called Multimodal Word Sense Translation. The aim of this new task is to correctly translate an ambiguous word given its context - an image and a sentence in the source language - while preserving its sense. Formally, if x is an ambiguous[1] word, v is the visual context which is an image pertaining to a particular sense of x, t is the textual context which is a source sentence describing the image in the source language, and Y is a collection of all possible lexical translations, then a probability distribution $p(y|x, v, t)$ could be used to get the most probable translation $\hat{y}$ of x which preserves its sense:

$$\hat{y} = \arg\max_{y \in Y} p(y|x, v, t) \tag{3.1}$$

For modelling $p(y|x, v, t)$, we need a labelled dataset of 4-tuples of the form:

$$\{(x_i, y_i, t_i, v_i)\}_{i=1}^{n} \tag{3.2}$$

where n is the size of the dataset and x_i, y_i, t_i, v_i are the ambiguous word, its sense preserving lexical translation, source sentence and image respectively. We created such a dataset and it is available for use and analysis at `github.com/sheffieldnlp/mlt`. Our language resource has several potential uses including evaluation of Word Sense Disambiguation capabilities of both, text-only and multimodal Machine Translation systems which we present towards the end of this chapter.

3.1 Creating the Dataset

To create our dataset, we made use of the already existing Multi30K dataset (Elliott et al. 2016, Specia et al. 2016, Elliott et al. 2017, Barrault et al. 2018) (see Table 2.2), an extension of the Flickr30K dataset (Young et al. 2014), which consists samples of the form (v_i, t_i, r_i)

[1] We use the term 'ambiguous' for those words in the source language that have multiple lexical translations in the target language in a given parallel corpus, loosely representing different 'senses' of the word in that corpus.

where v_i is a visual context (an image), t_i is a textual context (description of the image in the source language English) and r_i is a reference translation of the description in the target language (German or French or Czech) by human translators (i is an integer index representing a particular sample in the dataset). From this sentence-level dataset, we extracted the ambiguous words and their lexical translations using the following steps:

Pre-processing → Word Alignment → Automatic Filtering → Human Filtering

3.1.1 Pre-processing

Sentences in all languages were lowercased and tokenized using scripts from the Moses toolkit[2] (Koehn et al. 2007). German sentences, which may contain compound words like 'sonnenblumenkerne' (sunflower seeds), were split or decompounded using a pre-trained model of SEmantic COmpound Splitter (SECOS)[3] (Riedl & Biemann 2016). In the above example, 'sonnenblumenkerne' was decompounded to 'sonne blume kerne' corresponding to 'sun flower seed'. Even 'sonnenblume' corresponding to 'sunflower' is splitted to 'sonne blume' (sun flower). This is one caveat where decompounding may split a German word many times.

Also, since we are not interested in distinguishing morphological variants of the words, we lemmatized all sentences in the respective languages, which reduced the vocabulary size and led to better word alignment in the later step. For English, German and French, we used Ahmet Aker's Part Of Speech Tagger and Lemmatizer toolkit[4] which is based on the Helsinki Finite-State Transducer Technology (HFST) (Linden et al. 2013) and 'word-lemma' dictionaries. For Czech, we used the MorphoDiTa toolkit[5] (Straková et al. 2014). It was later observed that many words in all the languages did not get lemmatized because the Finite-State Transducer had not returned any lemma suggestions and these words were not found in the 'word-lemma dictionary' used by the Lemmatizers.

3.1.2 Word Alignment

In a parallel corpus, word alignment refers to aligning words in the source sentence to the words in the target sentence which may mean the same or play the same role as depicted in Figure 3.1. In Statistical Machine Translation, word alignment models are commonly used as important sub-components in the noisy channel formulation (see Equation 2.6).

After the pre-processing step, we aligned the word tokens in the Multi30K parallel corpus using Fast Align[6] (Dyer et al. 2013). We used Fast Align as compared to Giza++ (Och & Ney 2003) because it is more recent, faster and also because phrase based Statistical Machine Translation systems trained on Fast Align were found to perform better compared to those trained on Giza++ alignment model (Dyer et al. 2013). Fast Align generates asymmetric word alignments on a parallel corpus depending on which language in the parallel corpus is treated as the source language. For instance, in English-Czech parallel corpus, if English is treated as the source language (and Czech is the target language) then we get an alignment

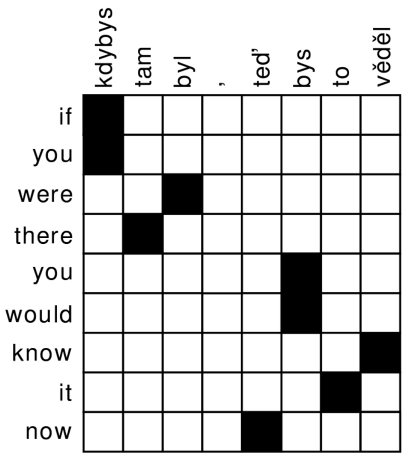
[2] https:/github.com/moses-smt/mosesdecoder

[3] https://github.com/riedlma/SECOS

[4] http://staffwww.dcs.shef.ac.uk/people/A.Aker/activityNLPProjects.html

[5] http://ufal.mff.cuni.cz/morphodita

[6] https://github.com/clab/fast_align



	kdybys	tam	byl	,	ted’	bys	to	věděl
if	■							
you	■							
were			■					
there		■						
you						■		
would						■		
know								■
it							■	
now					■			

Figure 3.1: *An example of word alignment. The words in the English sentence "if you were there you would know it now" and those in the Czech sentence "kdybys tam byl, ted' bys to věděl" are aligned represented by the black squares. For example, both "you" and "would" in English are aligned to "bys" in Czech.*

which is different from the alignment obtained when Czech is treated as the source language. In the example in figure 3.1, when English is the source language then the English word "it" is aligned to the Czech word "to", but when Czech is the source language then the Czech word "to" is aligned to the English word "would". In our case, we generated both alignments which we call - 'forward alignment' where English is treated as the source language, and 'reverse alignment' where German or French or Czech is treated as the source language. To learn better word alignments, we trained Fast Align models on a larger corpus comprising of the Europarl parallel corpus[7] Koehn (2005) in addition to the Multi30K parallel corpus for the English-German, English-French and English-Czech language pairs separately. The Europarl corpus was also pre-processed using the same pre-processing steps as indicated in Section 3.1.1 before word alignment.

3.1.3 Automatic Filtering

In this step we removed all the word alignments which have stop words[8] in either the source language or the target language. For English, French and German, we used the stop words list from Natural Language ToolKit (NLTK) (Loper & Bird 2002). For Czech, we used the Google stop-words dataset[9]. Next, we selected only those word alignments which were found in both the 'forward alignment' and the 'reverse alignment' directions. All other alignments were removed. In addition, we filtered out the alignments between words with different Part-Of-Speech[10] (POS) tags. For English, German and French we used Ahmet Aker's POS Tagger

[7] `http://www.statmt.org/europarl/`

[8] Stop words refer to words or tokens which can be removed from a dataset for a particular task because these do not add any significant value in solving that task. Most commonly, words in English like 'the', 'a', 'is', etcetera are not useful for many tasks so we remove them.

[9] `https://code.google.com/p/stop-words/`

[10] Part Of Speech tags refer to syntactic category of a word like Noun, Adjective, Verb, etcetera for English. We aligned Part Of Speech tags across languages.

(see footnote 4) and for Czech we used MorphoDiTa (see footnote 5). Next, we removed all those English words that were aligned to a single word in the target language across the entire Multi30K corpus because these are considered to be unambiguous words in English. This way we retained in our dataset only the potentially ambiguous English words, i.e. those aligned to multiple words in the target language. These retained alignments were converted into a dictionary format where 'Keys' are the potentially ambiguous English words and 'Values' are all the words in the target language that got aligned to it by the Fast Align model. For instance, in English-French language pair we have examples like:

four → *quart, quartequatre*

woods → *forêt, bois*

western → *occidental, western*

hat → *casque, casquette, chapeau, haut, bonnet, couvre, képi, béret*

A dictionary for each target language (German, French and Czech) from the word alignments of each language pair was built independently, i.e. one for English-German, one for English-French and one for English-Czech.

3.1.4 Human Filtering of Dictionaries

Finally, each dictionary (English-German, English-French and English-Czech) obtained from the Automatic Filtering step were given to human annotators for a final inspection and filtering. Human annotators were native speakers of the target languages (French or German or Czech) who were also fluent in English. They were asked to,

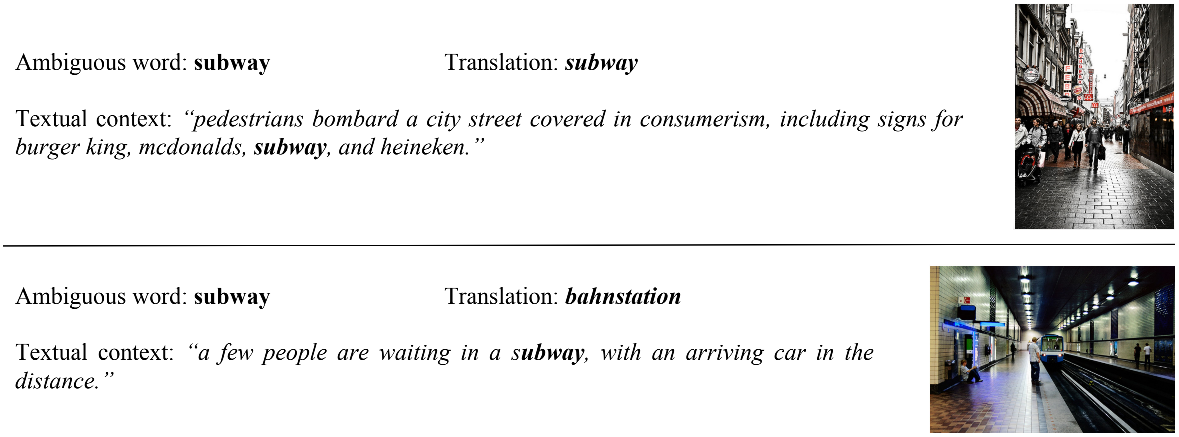
1. Filter out instances which they believed did not have multiple senses.
   For example, 'western → *occidental, western*'

2. Filter out target words (Values) which they believed are not translations of the source word (Key) in any context.
   For example, in 'hat → *casque, casquette, chapeau, haut, bonnet, couvre, képi, béret*', the French word *haut* is not a translation of the English word *hat* in any context so we removed it from the dictionary.

The annotators were given the freedom to use any other resource, such as bilingual dictionaries, existing translation tools, etcetera that may help them filter the dictionaries. We would like to point out, while several annotators worked on cleaning the dictionaries, each entry in the dictionary was cleaned by a lone annotator. Hence, we could not measure inter-annotator agreement. Also, it is important to note that ambiguity is a subjective concept and different annotators may disagree as to which words they consider are ambiguous. For example, some consider the English word 'white' is unambiguous while many others consider it to be ambiguous where one sense refers to white as the colour and the other sense refers to white as the race. We gave our annotators the freedom to decide which words they thought were ambiguous and which they thought were not.

After the final filtering and inspection of the dictionaries, for each (Key, Value) pair in the dictionaries we retrieved the visual and textual contexts (image v and its English description

t respectively) from the Multi30K dataset by searching for samples which have the Key in the English description t and the Value in the reference translation r. This way, we got 5-tuples of the form (x, y, v, t, r) where x is the Key which is the ambiguous word, y is the Value which is the sense-preserving translation of x. For our task of Multimodal Word Sense Translation, we do not need the reference translation so we can ignore r from the dataset. A couple of examples are shown below in Figure 3.2.



Ambiguous word: **subway** Translation: ***subway***

Textual context: *"pedestrians bombard a city street covered in consumerism, including signs for burger king, mcdonalds,* ***subway****, and heineken."*

Ambiguous word: **subway** Translation: ***bahnstation***

Textual context: *"a few people are waiting in a* ***subway****, with an arriving car in the distance."*

Figure 3.2: *Samples from the dataset for Multimodal Word Sense Translation. The ambiguous word 'subway' has two different sense translations - 'subway' (brand) and 'bahnstation' (metro train station) - according to the textual and visual contexts.*

3.1.5 Additional Human Filtering for the 2018 Test Set

The above procedure from pre-processing (section 3.1.1) to human filtering of dictionaries (section 3.1.4) was performed on the Train, Validation, Test 2016, Test 2017 Flickr and Test 2017 MSCOCO datasets of the Multi30K dataset taken as a whole (see Table 2.2). The Test 2018 dataset of Multi30K, which consists 1071 (v, t, r) parallel samples, was not available at the time when we created our dictionaries. So, for Test 2018, we retrieved the test instances, 5-tuples of the form (x, y, v, t, r), using the existing dictionaries created from the earlier datasets. We used the same string matching approach, i.e., for every (Key, Value) pair (x, y) in our dictionary we extracted samples (v, t, r) from Test 2018 Multi30K such that t contains the Key x and r contains the Value y.

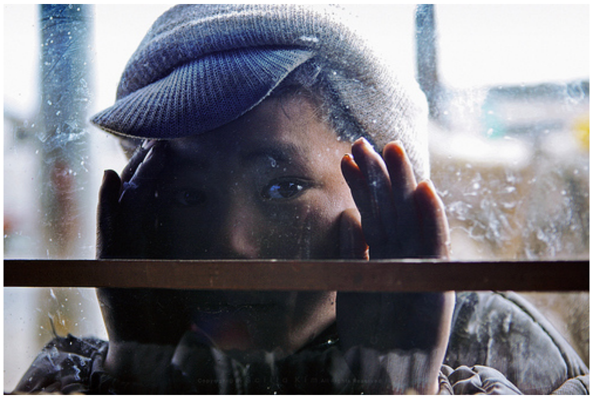
Next, from these test samples, the English description t together with the ambiguous word x and the set y^x of all possible Word Sense Translation candidates of the ambiguous word x were provided to human annotators who were bilingual speakers of both English and the target language under consideration (German or French or Czech). The corresponding image v was also provided but not explicitly shown to the annotators. They had the option to look at the image if they have to and specify that they used the image. Annotators for Czech, however, did not see the image at all when filtering the Test 2018 set.

The objective for the annotators was to select those sense-preserving translation candidates from y^x which they thought conform to both the English description and the corresponding image. In other words, they had to filter out those translation candidates that do not conform to either the English description or the image while having the options (a) to look at the image (if they think the visual context is needed to make a decision) or (b) ignore it completely (if they think the visual context is not needed to decide). If they selected all available translation candidates (i.e. did not filter out any single translation candidate) then those examples were removed. Also, the other extreme where they filtered out every translation candidate were also removed. They had the option to add their own sense-preserving translations conforming to the context if they wished to do so. We demonstrate the additional human filtering process with the following example,

We showed our annotator, who was a native speaker of German and also fluent in English, the following,

Ambiguous word: *hat*
Textual context: *a cute boy with his hat looking out of a window.*
Sense Translation Candidates: *kappe, mütze, hüten, kopf, kopfbedeckung, kopfbedeckungen, hut, helm, hüte, helmen, mützen*

Then the annotator was asked *"Do you need the corresponding image to know the context better in order to decide the correct sense-preserving translations of the ambiguous word?"* In this instance the annotator answered 'yes' and proceeded to look at the image in Figure 3.3.

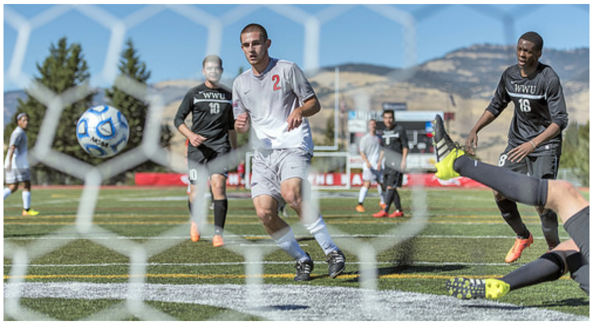


Ambiguous word: hat
Textual context: a cute boy with his **hat** looking out of a window.
Translation candidates: {kappe, mütze, hüten, kopf, kopfbedeckung, kopfbedeckungen, hut, helm, hüte, helmen, mützen}

Figure 3.3: *Example of Additional Human Filtering for the 2018 test set where image was used. Human annotator opted to look at the image to decide the set of correct sense-preserving translations of the ambiguous word* hat. *The annotator selected* {kappe, mütze, mützen} *as the set of correct sense-preserving translations from the set of translation candidates.*

After looking at the image, the annotator selected *kappe, mütze, mützen* as the correct sense-preserving translations of *hat* in the given context. We noted down that the image was seen to make a decision in this case. Next, we also checked if the reference translation r of the source sentence contained any of the selected sense-preserving translations. In this case, the reference translation is *"ein süß jung mit mütze blicken aus einem fenster"* which contains *mütze*.

In many other instances, like in Figure 3.4, the annotator answered 'no' and did the filtering without looking at the image because the annotator did not think the image was needed.



Ambiguous word: shot
Textual context: a soccer player in gray making a successful **shot** on goal in a soccer game.
Translation candidates: {aufnahme, bild, foto, schuss, schnäpse, schnaps, schießt, schießen, bowlingwurf, wurf}

Figure 3.4: *Example of Additional human filtering for the 2018 test set where image was not used. Human annotator did not opt to look at the image to decide the set of correct sense-preserving translations of the ambiguous word* shot. *The annotator selected* {schuss} *as the set of correct sense-preserving translations from the set of translation candidates.*

3.2 Analysis of the Dataset

The statistics of the dataset we created, excluding the 2018 Test set which underwent additional human filtering, is summarized in Table 3.1. Among several statistics mentioned in the table, we have something called the Averaged Skewness Ratio (ASR) which we will define in Section 3.2.1. The rest of the statistics are self-explanatory. We observe the various statistics of the English-Czech version of our dataset are much higher than the English-German and the English-French versions of the dataset. This is because the annotation was much more lenient in the 'Human Filtering of Dictionaries' step for English-Czech. The Czech annotators did not filter out instances which do not have multiple senses (see step 1 in Section 3.1.4). They only focused on ensuring the Values in the dictionary are valid translations of their corresponding Key. Therefore, unambiguous English words which have multiple different translations but with the same sense remained in the English-Czech dictionary and this resulted in a larger dictionary of ambiguous words and its sense translation. As a result of a larger dictionary, more samples were retrieved from the Multi30K dataset. We regard

Statistic	EnDe	EnFr	EnCz
Total number of samples	53868	44779	82096
Number of unique ambiguous words	745	661	1067
Average number of samples per unique ambiguous word	72.3	67.7	76.9
Average # of samples per sense translation	17.6	22.6	15.1
Average # of ambiguous words per sentence	1.6	1.3	2.5
Average # of ambiguous words per hundred words	15	12	24
Average # of Translation Candidates Per Ambiguous word (TCPA)	4.1	3.0	5.1
Averaged Skewness Ratio (ASR) (see Section 3.2.1)	1.8	1.6	1.9

Table 3.1: *Statistics of the Dataset for Multimodal Word Sense Translation excluding the 2018 Test set which went through additional human filtering. There are three versions of the dataset for each language pair. For each language pair, we have samples which are 5-tuples consisting an ambiguous word in English, its sense-preserving lexical translations in the target language, its textual context, its visual context and the reference translation of the textual context in the target language.*

English-Czech version of our dataset noisier than the other language pairs.

We can split our dataset further into Train, Validation, Test 2016, Test 2017 Flickr and Test 2017 MSCOCO versions using the same splits of the original Multi30K dataset. All we have to do is check from which version of the Multi30K dataset was the contextual information (v, t, r) retrieved for the given Key-Value pair (x, y) of ambiguous word and its lexical translation. The Test 2018 version of our dataset is different from the rest of the dataset because:

1. The Test 2018 version of Multi30K dataset was not used in creating the dictionaries of ambiguous words and its sense-preserving translations. However, we used (Key, Value) pairs from those dictionaries to retrieve contextual information (v, t, r) from the Test 2018 version of Multi30K.

2. The Test 2018 version of our dataset underwent additional human filtering.

3. Multiple lexical translations are considered to be correct unlike in the rest of the dataset where only one lexical translation is considered to be correct. In other words, y is a set of several labels and not just one label unlike before.

The statistics of the Test 2018 version of our dataset for Multimodal Word Sense Translation is summarized in Table 3.2. We observed that the number of English-Czech samples is significantly lower compared to other language pairs, unlike in Table 3.1. This is because the annotators tried to compensate for the noisier English-Czech version of our dataset and ended up aggresively filtering out samples in the Test 2018. Average number of acceptable sense translations per sample is the average size of the set y of labels. Two interesting new statistics we have in Test 2018 version of our dataset are (a) Number of samples where image was opted for and (b) Average number of visually ambiguous words per hundred words which is defined in Section 3.2.2.

We will now analyse our datasets, especially the additional human filtering which involved annotators looking at images, in further detail in the following subsections.

Statistic	EnDe	EnFr	EnCz
Total number of samples	358	438	140
Number of unique ambiguous words	38	70	29
Average number of samples per unique ambiguous word	9.4	6.3	4.8
Average # of translations candidates per ambiguous word (TCPA)	4.1	3.0	5.1
Average # of acceptable sense translations per sample	2.6	1.5	3.3
Average # of ambiguous words per hundred words	2.6	2.9	1.4
Number of samples where image was opted	111 (31%)	72 (16%)	-
Average # of visually ambiguous words per hundred words	0.8	0.5	-

Table 3.2: *Statistics of the Test 2018 Dataset for Multimodal Word Sense Translation which went through additional human filtering. There are three versions of the dataset for each language pair. For each language pair, we have samples which are 5-tuples consisting an ambiguous word in English, its sense-preserving lexical translations in the target language, its textual context, its visual context and the reference translation of the textual context in the target language.*

3.2.1 Skewed Distributions of Lexical Translations

A key aspect of our dataset worth noting is the skewed distribution over the lexical translation candidates for a given ambiguous word. For instance, the English word *woods* has two possible lexical translations in French in our dataset, *forêt* (forest) and *bois* (wood). Ideally, we would want both these lexical translations to occur equal number of times (uniform distribution) but in reality the distribution is skewed - *bois* occurs 79 times (we call it the Most Frequent Word Sense Translation) while *forêt* occurs only 16 times. Another example is the English word *lean* which has the following translations in German in our dataset - *lehnen* (to be leaning), *schlank* (slim), *stützen* (support), and *beugen* (bend). In our dataset, *lehnen* is the Most Frequent Word Sense Translation of *lean* which occurs 137 times while the rest of the translation candidates combined occur only 16 times. Such a skewed distribution over translation candidates makes words like *woods* and *lean* virtually unambiguous (or less ambiguous) compared to the cases when the distribution is more uniform over the translations like the word *pack*. The English word *pack* has the following translations in German in our dataset - *gruppe* (group), *rudel* (herd), *packen* (to pack) and *packung* (box or packet). In our dataset, we have four samples of *gruppe*, and three each of *rudel*, *packen* and *packung*. This is a much more uniform distribution over translation candidates which makes it difficult to disambiguate. We would like to further quantify this aspect of Word Sense Translation below.

For a better understanding of the skewness of the distributions over translation cadidates, we define a simple heuristic called Skewness Ratio (SR) of a word as the ratio of count of the word to the count of its most frequent translation. Formally, let x be an ambiguous word with n different translation candidates $y_1, y_2, ..., y_n$. Let $freq(y_i|x)$ denote the number of times the word y_i occurs as a translation of x in the training set. Also, without loss of generality, arrange the translations in the decreasing order of frequency, i.e. $freq(y_1|x) > freq(y_2|x) > ... > freq(y_n|x)$. Then we define the Skewness Ration of x as,

$$\mathrm{SR}(x) = \frac{\sum_{i=1}^{n} freq(y_i|x)}{freq(y_1|x)} = \frac{freq(x)}{freq(y_1|x)} \tag{3.3}$$

We note, our definition of Skewness Ratio is similar to the inverse of 'Average Time-anchored Relative Frequency of Usage' metric defined in Ilievski et al. (2016) which is used to assess potential bias of meaning dominance. We have formulated Skewness Ratio this way because we want higher value to reflect a more uniform distribution over translation candidates.

For the examples of *wood*, *lean* and *pack* mentioned earlier, we compute the Skewness Ratios as follows:

$$\mathrm{SR}(\text{woods}) = \frac{freq(\text{woods})}{freq(\text{bois}|\text{woods})} = \frac{16+79}{79} = 1.20 \tag{3.4}$$

$$\mathrm{SR}(\text{lean}) = \frac{freq(\text{lean})}{freq(\text{lehnen}|\text{lean})} = \frac{16+137}{137} = 1.11 \tag{3.5}$$

$$\mathrm{SR}(\text{pack}) = \frac{freq(\text{pack})}{freq(\text{gruppe}|\text{pack})} = \frac{4+3+3+3}{4} = 3.25 \tag{3.6}$$

The skewness ratio of both *lean* and *woods* is close to 1 which suggests both have an extremely skewed distribution over their lexical translations. Out of the two, *woods* has less skewed distribution over lexical translations than *lean* and hence, would appear slightly more difficult to disambiguate. On the other hand, the skewness ratio of *pack* is 3.25 which is closer to 4 - its total number of lexical translation candidates. This suggests, it is far more uniformly distributed across its Word Sense Translations and hence most difficult to disambiguate. To quantify the skewness over distributions of translations for the entire dataset, we can compute the Average of all the Skewness Ratios (ASR) averaged over all the ambiguous words in our dataset. Formally, if our dataset D has a total of n unique ambiguous words $x_1, x_2, ..., x_n$, then

$$\mathrm{ASR}(D) = \frac{\sum_{i=1}^{n} \mathrm{SR}(x_i)}{n} \tag{3.7}$$

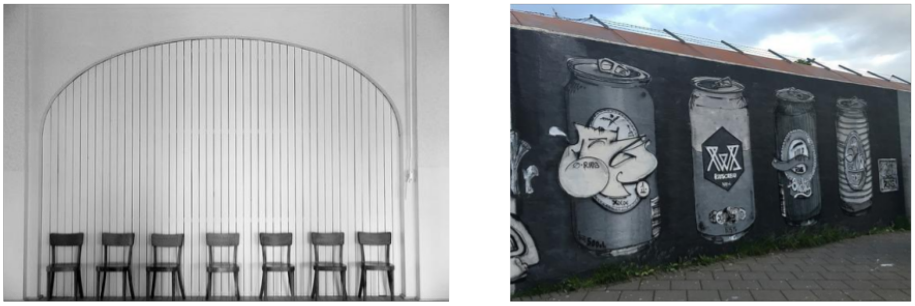
ASR will be a number between 1 and the average number of sense Translation Condidates Per Ambiguous word (TCPA). If it is closer to 1 then it means that, in the dataset, the distribution over lexical translations is highly skewed. If it is closer to TCPA, then the distribution over lexical translation is more uniform. The ASR and TCPA for our dataset has been mentioned in Table 3.1. For English-German, ASR of 1.8 is closer to 1 as compared to 4.1 which is the TCPA. Similar numbers are seen for English-French and English-Czech versions. This suggests that our dataset is highly skewed and that the Most Frequent Word Sense Translation appears far too often compared to other translation candidates. This makes our dataset challenging because, as noted in Postma et al. (2016) for Word Sense Disambiguation, any model for Multimodal Word Sense Translation will find it difficult to beat the simple baseline model which just returns the Most Frequent Word Sense Translation.

3.2.2 When Humans Find Images Useful

The additional human filtering for the 2018 Test set (see Section 3.1.5) allowed human annotators to consider the visual context when deciding which sense-preserving translation to select. This forms an interesting experiment to see when and how often humans feel the need to look at images when translating ambiguous words.

For English-German, the Test 2018 dataset consists of 358 instances of ambiguous words. In 111 (or 31%) of these instances, the annotators opted to look at the image. This is a promising result because it shows images could potentially be very useful for Multimodal Word Sense Transaltion in 31% of the samples. We shall call the ambiguous words in these 111 samples "visually ambiguous" because human annotator felt the need to look at the visual context in order to decide how to disambiguate it. Despite being a promising observation, it is important to note that from the perspective of Machine Translation this is actually a very small number because these 111 ambiguous words are spread in 1071 sentences consisting of 13,774 words in total. This means, we have found only 0.8% words which may directly benefit from the image. For English-French these numbers are even lower (see Table 3.1). Although the sample size is small, these numbers help us understand the scope of using images for Multimodal Word Sense Translation and Multimodal Machine Translation. Next we look at the visually ambiguous words qualitatively.

Ambiguous words for which human annotators opted to look at the image include *young, pool, hat, coat, field, wall*, suggesting textual context is not sufficient for such words and visual context is essential to make a correct lexical choice in the translation. Consider the word *wall*. It has two lexical translations in German *mauer* and *wand*. *Mauer* refers to the wall from the outside, while *wand* refers to the wall from the inside (see Figure 3.5). A simple sentence like "We saw a graffiti on a wall" does not have any evidence that the graffiti is on



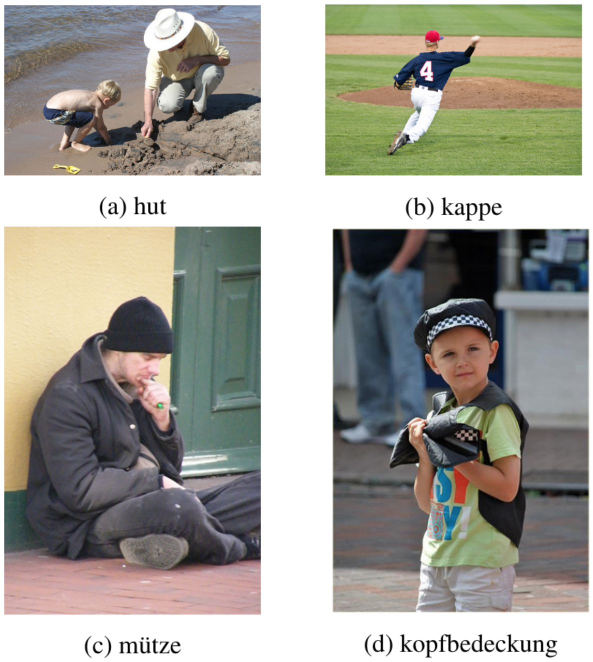
Wand
A wall seen from the inside

Mauer
A wall seen from the outside

Figure 3.5: *Different sense-preserving lexical translation of the ambiguous English word 'Wall' into German words 'Wand' or 'Mauer'. Textual context is often not sufficient to determine the correct sense translation of 'wall' and visual context is often necessary. Therefore, we call it 'visually ambiguous'. Also, 'wall' is not considered ambiguous in English while it is seen ambiguous when translating into German. Therefore, we may consider it to be an example of 'transfer ambiguity'. The different sense translations of wall are closely related to each other. Therefore, we may also consider it to be an example of polysemy.*

the external side of the wall or the internal side. This distinction, which is needed for an accurate translation of the sentence into German, is possible in most cases only when the visual context is provided. We therefore consider the word ‘wall’ to be ‘visually ambiguous’. Also, notice that the word ‘wall’ is not ambiguous from the perspective of the source language English. It is only from the perspective of German that it is ambiguous. Recall from Section 2.1.3, transfer ambiguities refer to words which are not ambiguous from the perspective of the source language but these are ambiguous from the perspective of the target language. We may therefore consider ‘wall’ to be an example of ‘transfer ambiguity’. Finally, the different senses of ‘wall’ (*mauer* and *wand*) are closely related to each other because it is the same wall but seen from different sides. We may therefore consider it to be an example of ‘polysemy’ (see Section 2.1.3).

Two particular words in English from our dataset which have been repeatedly considered visually ambiguous by both German and French annotators are the words *hat* and *coat*. These are clothing products which are closely tied to the cultures around the world which have different names for various clothes they wear. In English, *hat* and *coat* are not ambiguous but they are ambiguous when translated into German or French (see Figure 3.6). The translation


(a) hut

(b) kappe

(c) mütze

(d) kopfbedeckung

Figure 3.6: *Different kinds of ‘hats’ translated into German differently based on the visual context. Textual context is often not sufficient to determine the correct sense translation of ‘hat’ and visual context is often necessary. Therefore, we call it ‘visually ambiguous’. Also, ‘hat’ is not considered ambiguous in English while it is seen ambiguous when translating into German and French. Therefore, we may consider it to be an example of ‘transfer ambiguity’. The different sense translations of hat are closely related to each other. Therefore, we may also consider it to be an example of polysemy.*

varies depending on the type of *hat* or the type of *coat*. Therefore, like *wall*, we consider *hat* and *coat* as 'transfer ambiguities'. We may also consider these ambiguous words to be 'polysemous'. For example, in the case of *hat*, all the different sense translations / variants of hats have a common underlying concept which is"something you wear on your head". The different senses of *hat* are depicted in Figure 3.6. These are (a) *hut* which refers to hats with edges/extensions coming off from all sides and usually worn in summer, (b) *kappe* which refers to the modern caps with shades extending out from front side only, usually worn at sporting events, (c) *mütze* which refers to differently designed hats usually worn in winter and (d) *kopfbedeckung* which means a headgear which could refer to any kind of object worn on the head.

Ambiguous words where human annotators ignored the image include *area, fall, watch, walk,* etcetera, suggesting the textual context is often sufficient to identify the correct translation. These are usually 'category ambiguities'. Recall from section 2.1.3, category ambiguity refers to words with the same surface form but different category like different Part-of-Speech tag. For example, *watch* can be a verb (to see) or noun (instrument to measure time). Category ambiguities can be disambiguated using other means like Part-of-Speech Tagging models and need not use the image to do that. Human annotators in our experiment also chose not look at the image for these cases. We may therefore call such words 'textually ambiguous' because the textual context is often sufficient and the visual context is not needed to resolve such ambiguities.

Next, we filtered our English-German and English-French test datasets (Test 2016, Test 2017 Flickr, Test 2017 MSCOCO, Test 2018) further to exclude the textually ambiguous words identified in the additional human filtering above. This way, we got subsets of our test sets which contain (1) visually ambiguous words and (2) other ambiguous words whose type of ambiguity (visual or textual) had not been identified because these were not part of the Human experiment. We shall call these subsets of the test sets as 'Visually Ambiguous subset'. The number of samples in the original test sets and the Visually Ambiguous subsets are given in the following Table 3.3. We shall use these test sets in our Multimodal Word Sense Translation experiments.

Language Pair	Test 2016	Test 2017 Flickr	Test 2017 MSCOCO	Test 2018
		Original Test sets		
English-German	1004	880	381	358
English-French	864	929	441	438
		Visually Ambiguous subsets of the test sets		
English-German	534	476	330	111
English-French	287	256	173	72

Table 3.3: *The sample size of the test sets used for Multimodal Word Sense Translation. The Visually Ambiguous subset contains visually ambiguous words identified in the additional human filtering of Test 2018 dataset in section 3.1.5 and other ambiguous words which are not identified to be textually ambiguous. Since Test 2016, Test 2017 Flickr and Test 2017 MSCOCO were not part of the additional filtering, therefore their sizes are much higher than expected.*

Manual inspection of visually ambiguous words and textually ambiguous words of our dataset suggests visual context may benefit transfer ambiguities and polysemes more as compared to other kinds of lexical ambiguities. This may also indicate why traditional monolingual Word Sense Disambiguation have not been significantly useful for Machine Translation, simply because the main problem is 'transfer ambiguities' (words unambiguous from the perspective of source language, but ambiguous from the perspective of target language) which cannot be resolved using monolingual Word Sense Disambiguation models.

3.3 Evaluating Multimodal Word Sense Translation capabilities of Multimodal Machine Translation Systems

We can use the dataset for Multimodal Word Sense Translation to evaluate the disambiguation capabilities of both Monomodal and Multimodal Machine Translation systems. Consider a sample from our dataset which is of the form (x, y, t, v) where x is an ambiguous word and y is its sense-preserving lexical translation conforming to both - the textual context t and the visual context v. Let M be a Monomodal or Multimodal Machine Translation system which can read the textual context t and the visual context v as inputs to generate a translation which we will refer to as $M(x, t, v)$. If the system is monomodal then it only takes t as input and ignores v. A straightforward evaluation strategy is to simply check if the correct sense-preserving lexical translation y is also found in the system's output $M(x, t, v)$[11]. If it is found, then we say that our system disambiguated the ambiguous word x correctly from the given context.

An important caveat is that we assume y is the only correct sense-preserving lexical translation of x which may not be true. For many examples, some other word z which may be a sense-preserving synonym of y may also be considered to be a correct sense-preserving lexical translation of x for the given context (t and v). So if the system generates a translation which consists z instead of y then it will end up being wrongly ignored. We addressed this problem to some extent in our pre-processing steps (Section 3.1.1) by lemmatizing. So if y is found in lemmatized $M(x, t, v)$ then M will be considered to disambiguate x correctly. In addition, with the additional human filtering introduced in the Test 2018 dataset (Section 3.1.5), the single label y is replaced by a set $y = \{y^1, y^2, ..., y^k\}$ of acceptable sense-preserving translations of x for the given context. In this situation, if any $y^i \in y$ is found in the system generated translation $M(x, t, v)$ then we consider the system has disambiguated x correctly. On the other hand, if every $y^i \in y$, is not found in $M(x, t, v)$ then we consider the system to have failed to disambiguate the ambiguous word x. Formally, this could be a function called 'Check(M, x, t, v, y)' defined as follows:

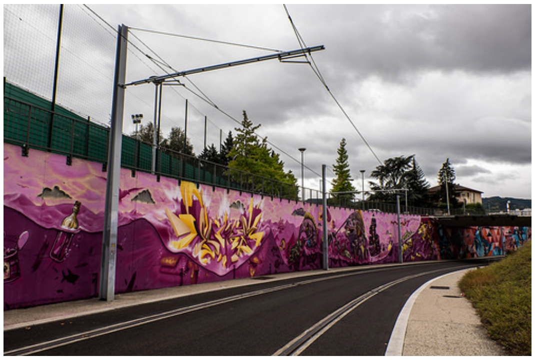
$$\text{Check}(M, x, t, v, y) = \begin{cases} 1, & \text{if } \exists y' \in y \text{ such that } y' \in M(x, t, v) \\ 0, & \text{if } \forall y' \in y, y' \notin M(x, t, v) \end{cases} \tag{3.8}$$

Here is an illustration from the Test 2018 version of our dataset for Multimodal Word Sense Translation (see Figure 3.7). We show Google Translate generates an output[12] which translates the ambiguous word 'wall' as 'wand' referring to the wall as being seen from inside

[11] For consistency, the system's output should undergo the same steps in Section 3.1.1 which was used to create our dataset of ambiguous words and its sense-preserving lexical translations

[12] As of September 2019 on Google Translate: `https://translate.google.co.uk/`

which is a wrong sense translation for the given context as the wall is being seen from the outside which is evident in the image. Ofcourse, Google Translate is a text-only system but it could have, perhaps, benefited from the visual context.



Ambiguous word x: wall
Textual context t: colorful purple graffiti art covers a wall alongside a curved road.
Acceptable senese-preserving lexical translations y: {mauer, mauern}

Google Translate output $M_{Google}(x, t)$:
bunte lila graffitikunst bedeckt eine wand entlang einer gekrümmten straße.

Reference translation of source Sentence $r = M_{ref}(x, t, v)$:
farben froh violette graffiti bedecken eine mauer entlang einer straßen kurve.

Figure 3.7: *Evaluating Word Sense Translation capabilities of both Monomodal and Multimodal Machine Translation systems. Using the function* Check *defined in the equation 3.8, we get Check*$(M_{Google}, x, t) = 0$*, i.e. Google Translate failed to translate the correct sense of the ambiguous word 'wall'. It generated the word 'wand' which refers to a wall seen from the inside.*

To measure the overall Word Sense Translation performance of a system M over some version of our dataset $\{(x_i, y_i, v_i, t_i)\}_{i=1}^{n}$ for Multimodal Word Sense Translation (see Equation 3.2), we can simply count the number of times the system translated the ambiguous words correctly using the 'Check' function in Equation 3.8 and compute its accuracy. We call this accuracy measure the 'sense-preserving Lexical Translation Accuracy' (LTA) of a system. Formally,

$$\text{LTA}(M) = \frac{\sum_{i=1}^{n} \text{Check}(M, x_i, t_i, v_i, y_i)}{n} \tag{3.9}$$

We measured Lexical Translation Accuracy of all the systems that were submitted to the Multimodal Machine Translation shared task of 2017 and 2018 (Elliott et al. 2017, Barrault et al. 2018). A total of 97 Monomodal and Multimodal Machine Translation systems were evaluated. All the evaluation results can be found in the tables in Appendices B.1, B.2 and B.3. The systems being referred to by the abbreviated names in these tables can be found on the official website[13] of the Multimodal Machine Translation shared tasks.

[13] `www.statmt.org/wmt18/multimodal-task.html` and `www.statmt.org/wmt17/multimodal-task.html`

We would like to point out that our dataset has many ambiguous words with skewed distribution over their lexical translation candidates. Many such words, like *woods* and *lean* (see Equations 3.4 and 3.5), which have Skewness Ratio close to 1 can be regarded as virtually unambiguous (simply because the other sense translations are rare). To make our Test dataset challenging, we created a new dataset where we removed such words which have a Skewness Ratio less than or equal to 1.2 and retained only those words which have a more uniform distribution over their translation candidates. Multimodal Machine Translation systems were also evaluated on this challenging dataset of ambiguous words with Skewness Ratio > 1.2 and can be found in Appendix B.2. We observe that the Lexical Translation Accuracy performance of systems on this challenging test set is much lower than the original test set (see the difference in LTA scores of systems in Appendix B.2 with respect to their scores in Appendix B.1). For our future experiments in Chapter 3 and 4, we decided to select Skewness Ratio > 1.3 for our test sets to get rid of virtually unambiguous words and make it more challenging. However, please note that making the dataset challenging by selecting ambiguous words with higher Skewness Ratio scores comes at the cost of shrinking the dataset. In the case of selecting Skewness Ratio > 1.3, the test datasets shrunk by around 36%. But another way to look at it is, 36% of virtually unambiguous words were filtered out from the test sets.

Next, we compared our metric of sense-preserving Lexical Translation Accuracy with the other metrics which are generally used in evaluating the systems. Besides the standard Machine Translation metrics of BLEU, METEOR and TER (see Section 2.4.2), the submitted systems were also evaluated by human evaluators. Human scoring was carried out using bilingual Direct Assessment (Graham et al. 2017), where the assessors were asked to evaluate the semantic relatedness between the system outputs and the source sentence (not the reference translation) given the image. The assessors gave a sentence-level score between 0 and 100, where 0 indicates that the meaning of the source sentence is not preserved in the system output, and 100 means that the meaning is 'perfectly' preserved. The sentence-level scores were standardized according to each individual assessor's overall mean and standard deviation score. The overall Human score of a system was then computed as the mean of the standardized sentence-level scores over the test set. We note that, because human evaluation is costly, not all systems were evaluated by humans.

We observe that our 'sense-preserving Lexical Translation Accuracy' evaluation is consistent with other metrics and human scores. To measure the extent of this consistency, we computed the Pearson's Correlation Coefficient ρ_p (Pearson 1901, Benesty et al. 2009) (see Appendix A.10) and Spearman's Rank Correlation Coefficient ρ_s (Zar 2014) (see Appendix A.11) between our metric and METEOR metric (and Human scores where available). The correlation scores are presented in the following Table 3.4.

We observe that our metric, which evaluates a very specific aspect of translation, which is sense preservation in the translation of ambiguous words, positively correlates with both the METEOR metric and Human evaluation scores for Machine Translation. Ranking of the systems according to Lexical Translation Accuracy differs only slightly from the ranking using METEOR or human scores, and the top performing systems are often the same. These positive correlation scores validate the relevance of our metric. However, we would like to highlight certain caveats of using Lexical Translation Accuracy:

- Lexical Translation Accuracy is measured for only those sentences which have an am-

Dataset	Statistic	English-German	English-French	English-Czech	All-Czech
Test 2017 Flickr	ρ_s(LTA, METEOR)	0.94	0.93	-	-
	ρ_p(LTA, METEOR)	0.99	0.94	-	-
	ρ_s(LTA, Human)	0.90	0.54	-	-
	ρ_p(LTA, Human)	0.78	0.68	-	-
Test 2017 MSCOCO	ρ_s(LTA, METEOR)	0.80	0.95	-	-
	ρ_p(LTA, METEOR)	0.90	0.96	-	-
Test 2018	ρ_s(LTA, METEOR)	0.96	0.90	0.86	0.8
	ρ_p(LTA, METEOR)	0.92	0.88	0.84	0.78
	ρ_s(LTA, Human)	0.73	0.53	0.65	0.73
	ρ_p(LTA, Human)	0.64	0.59	0.60	0.66

Table 3.4: *Pearson's Correlation Coefficient ρ_p and Spearman's Rank Correlation Coefficient ρ_s between sense-preserving Lexical Translation Accuracy (LTA) and other metrics evaluating the systems submitted to the Multimodal Machine Translation shared tasks of 2017 and 2018. Note: All-Czech represents the task where Multimodal Machine Translation systems used inputs in English, French and German alongwith the image features to produce a translation in Czech. For more details, refer Barrault et al. (2018)*

biguous word. Sentences which do not have ambiguous words are ignored. On the other hand, METEOR, BLEU, TER and Human scores evaluate all sentences in a test set.

- Some sentences can have multiple ambiguous words which get evaluated by Lexical Translation Accuracy. In other words, Lexical Translation Accuracy is evaluating such sentences multiple times, one for each ambiguous word in it. However, the same sentence gets evaluated only once by METEOR, BLEU, TER and Human.

Finally, it is important to note that only a specific aspect of translation is being considered by Lexical Translation Accuracy. Therefore, we cannot consider it to be a primary metric to evaluate the performance of Multimodal Machine Translation systems. At best, Lexical Translation Accuracy can be regarded as a metric which supplements other metrics. It can be used for error analysis to identify instances where the system is disambiguating the ambiguous words incorrectly.

Next we explore if Lexical Translation Accuracy reveals any interesting trends in the Multimodal Machine Translation shared task. We found that, for teams which submitted text-only and multimodal variants of models, their multimodal versions seem to perform better at Lexical Translation Accuracy compared to their text-only counterparts. However, this trend is not visible using the METEOR, BLEU, or TER metrics. For example, the CUNI team (Helcl et al. 2018) submitted a Monomodal text-only Machine Translation model named 'CUNI_1_FLICKR_DE_NeuralMonkeyTextual_U' and Multimodal Machine Translation model named 'CUNI_1_FLICKR_DE_NeuralMonkeyImagination_U' (see Table B.9). Their multimodal model performs better at Lexical Translation Accuracy than their text-only model. On the other hand their text-only model performs better than multimodal model on other Machine Translation metrics BLEU, METEOR and TER. As mentioned in section 3.2.2, there are very few 'visually ambiguous words' spread across the Multi30K test sets. Changes made to these few ambiguous words will go unnoticed in metrics like BLEU, METEOR and TER due to their tiny presence. Hence, while Multimodal systems were performing better on LTA, this improvement was not seen by other metrics. We may therfore argue that our metric based on the dataset for Multimodal Word Sense Translation is a more fine-grained metric of evaluation of Machine Translation systems. Another important trend we observed is that the SHEF systems (see Appendix B.3), which will be presented in the next Chapter, that were built precisely to perform Multimodal Word Sense Translation, perform well on the LTA metric compared to other systems. It is important to note that these systems were not the best performing systems as per other metrics.

To summarise, we present our datset for Multimodal Word Sense Translation which can be used to make a fine-grained evaluation of sense translation in Multimodal Machine Translation systems. Next, we will develop supervised models for Multimodal Word Sense Translation using this dataset.

Chapter 4

Models for Multimodal Word Sense Translation

In the previous chapter, we created a dataset $\{(x_i, y_i, t_i, v_i)\}_{i=1}^{n}$ for Multimodal Word Sense Translation. Now, we will develop models for modelling $p(y|x, t, v)$ using the created data. Depending on what goes in as an input to the model, we can have several different variants of the task like,

- Word Sense Translation without any context. In other words, a distribution over sense translations for a given ambiguous word: $p(y|x)$
- Image-only Sense Label Prediction*: $p(y|v)$
- Text-only Sense Label Prediction*: $p(y|t)$
- Image-only Word Sense Translation: $p(y|x, v)$
- Text-only Word Sense Translation: $p(y|x, t)$
- Multimodal Sense Label Prediction*: $p(y|t, v)$
- Multimodal Word Sense Translation: $p(y|x, t, v)$
- Distribution over all different sense labels: $p(y)$

* refers to tasks where the ambiguous word is not specified. In such tasks, the model is directly predicting the sense label from the context without knowing the ambiguous word. Please note, in our dataset, the textual context t has the ambiguous word x in it. However, it is also important to note, many sentences may have multiple ambiguous words which makes $p(y|t)$ and $p(y|x, t)$ slightly different from each other.

4.1 Most Frequent Translation

Some straightforward baseline models can be extracted directly from the statistics of the dataset. We begin with the distribution of a sense label $p(y)$ over all possible sense labels of all ambiguous words combined together. This can be estimated simply as the ratio of frequency of a sense-label y over the size of the training set n. At test time, this model simply returns the label $\hat{y}$ which appears most number of times in the training set. Therefore, we call this the Most Frequent Label model (MFL) formulated below. Please note, this model completely disregards the ambiguous word x.

$$\text{MFL} = \hat{y} = \arg\max_{y \in Y} p(y) = \arg\max_{y \in Y} \text{freq}(y) \tag{4.1}$$

The Most Frequent Label for English-German is *wasser* (water), for English-French is *eau* (water) and for English-Czech is *košile* (Shirt; which is different from other two examples). The performance of this basic model across different test sets is presented in Table 4.1.

Model	Language Pair	Test 2016	Test 2017 Flickr	Test 2017 MSCOCO	Test 2018
Most Frequent Label	English-German	3.0	6.3	3.9	13.7
	English-French	5.2	7.8	3.4	0
	English-Czech	7.6	1.9	0	2.9
	Visually Ambiguous subsets of the test sets				
Most Frequent Label	English-German	3.6	6.8	4.0	17.2
	English-French	4.8	5.9	3.8	0

Table 4.1: *Most Frequent Label model. This model returns the most frequent sense label across all sense labels of all ambiguous words combined, completely disregarding the ambiguous word. Performance measured in Lexical Translation Accuracy (see Equation 3.9) as percentage between 0 and 100.*

We would like to highlight that despite being a dumb model with low performance, its performance is surprisingly higher than expected. This reflects the skewed nature of our dataset. In this case, we see skewness at two levels (a) Skewed distribution over ambiguous words and (b) Skewed distribution over its sense translations. In this case, *water* is among the most frequently occurring ambiguous words in our dataset and its sense-translation, *wasser* in German and *eau* in French, is its most frequent sense translation. At both levels (occurrence of *water* and occurrence of *wasser* or *eau* as its sense) the dataset is skewed. Another interesting point we would like to highlight is the impact of additional human filtering in the Test 2018 set (see Section 3.1.5). The German annotator chose to add *wasser* and remove several other words from the English-German Test 2018 dataset while the French annotator chose to remove *eau* from the English-French Test 2018 set. As a result, occurrence of *wasser* in the English-German Test 2018 set is higher than usual while *eau* does not appear in the English-French Test 2018 dataset at all. Therefore, Most Frequent Label model has a Lexical Translation Accuracy score of 13.7 in the English-German Test 2018 dataset and 0 in English-French Test 2018 dataset. Finally, we would also reiterate that the English-Czech dataset is noisy and therefore, more experiments were performed using only the English-German and English-French datasets.

Next, we look at Word Sense Translation without any context $p(y|x)$. Here the ambiguous word x is specified and we consider only the distribution over its sense translations. We denote the set of all possible lexical translations of x as Y_x. A simple way to estimate $p(y|x)$ is to take the ratio of frequency of sense translation label y over the frequency of the ambiguous word x in the training set. At test time, this model returns the sense $\hat{y}$ which appears most number of times as the lexical translation of x in the training set. We call this the Most Frequent Translation model (MFT) formulated below,

$$\text{MFT}(x) = \hat{y} = \arg\max_{y \in Y_x} p(y|x) = \arg\max_{y \in Y_x} (\frac{\text{freq}(y)}{\text{freq}(x)}) = \arg\max_{y \in Y_x} \text{freq}(y) \tag{4.2}$$

The performance of this basic model across different test sets is presented in Table 4.2. The high Lexical Translation Accuracy of this simple model reflects the skewed distribution over lexical translation candidates of a given ambiguous word. As pointed out in Postma et al. (2016), the Most Frequent Translation model will be extremely tough to beat because the most frequent sense translation label will dominate due to the skewness of the dataset.

Model	Language Pair	Test 2016	Test 2017 Flickr	Test 2017 MSCOCO	Test 2018
Most Frequent Translation	English-German	65.3	60.5	52.5	63.6
	English-French	77.7	77.3	67.1	75.1
	Visually Ambiguous subsets of the test sets				
Most Frequent Translation	English-German	65.6	60.9	54.0	65.3
	English-French	77.1	77.0	66.4	70.0

Table 4.2: *Most Frequent Translation model. This model returns the most frequent sense translation of a given ambiguous word. Performance measured in Lexical Translation Accuracy (see Equation 3.9) as percentage between 0 and 100.*

4.2 Other Simple Baseline Models

For some variants of the task, we tried and tested two simple baseline multi-class classification models: (a) k-Nearest Neighbours (kNN) and (b) single layer Feed-Forward Neural Network (FFNN). Formally, let f be a feature vector representing the input. Let Y be the set of all possible sense translation labels. Note, the total number of sense translation labels $|Y|$ is 3045 for English-German and 1970 for English-French. Let our training set of feature-label pairs be denoted as $\{(f_i, y_i)\}_{i=1}^{n}$.

Next, given a new feature f, in k-Nearest Neighbours we find k features nearest to f from the training set. Without loss of generality, let these be $\{(f_j, y_j)\}_{j=1}^{k}$, then the label $\hat{y}$ which is most frequent in this set is assigned as the label to the new feature f. We explored different values of k (3, 5, 10, 30, 50, 100, 500) and different distance metrics (see Appendix A.13). Then using grid search over validation set, i.e. to test the performance of k-Nearest Neighbour model for different values of k and different distance metrics over the validation set and then retrieve the best performing k and distance metric, we find the optimum value of k and the distance metric.

In the single softmax layered Feed-Forward Neural Network, which is depicted in Figure 4.1, we have a weight matrix W which is of the dimension $|f| \times |Y|$, and a bias vector b which is of the dimension $|Y|$. Then a new input feature f is assigned a label $\hat{y} = y_k$, where k corresponds to the k^{th} element in $W \cdot f + b$ which has the highest value. Formally,

$$k = \arg\max_{k \in \{1,2,\ldots,|Y|\}} \text{Softmax}(W \cdot f + b) \tag{4.3}$$

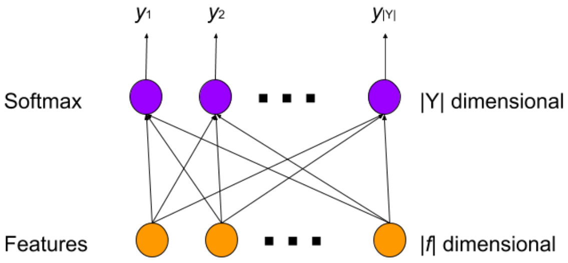


Figure 4.1: *Feed-Forward Neural Network as a baseline model.*

where Softmax is defined in Appendix A.12. In other words, this is a simple linear classifier. Softmax only converts the linear projection $W \cdot f + b$ into a probability distribution over the sense translation labels. The model parameters W and b are learnt by training the model on the training set using backpropagation algorithm and cross-entropy loss function (see Appendix A.14). For optimization, we use the Adam (Kingma & Ba 2014) algorithm with a learning rate set to 0.001 and batch size set to 32. Other hyperparameters include dropout, which is set to 0.3, and early stopping where training is stopped if model accuracy over the validation set does not improve for 30 epochs. Then the best performing model over the validation set is selected. Our Neural Network models are implemented and trained in the TensorFlow (Abadi et al. 2016) framework.

Consider the variant of our task which is Image-only Sense Label Prediction where we want to model $p(y|v)$. Here, we do not specify the ambiguous word or provide the textual context. We have to predict the sense label directly from the image features. We extracted image features of an image from the ResNet-50 Image classification model. More specifically, we extract Pooled features from the `pool5` layer of a ResNet-50 model trained on ImageNet. Then using these features as input, we train our k-Nearest Neighbours and Feed-Forward Neural Network classifiers. For k-Nearest Neighbours we got the best performance for k = 500 and Manhattan distance. The performance of these classifiers for Image-only Sense Label prediction are presented in Table 4.3.

We observe the k-Nearest Neighbours model performs extremely poorly even for the highest value of k = 500. It is unable to outperform even the Most Frequent Label (see Table 4.1). The ones that this model gets correct are mostly the Most Frequent Label. We speculate that the performance will keep increasing marginally if we keep increasing k to the size of the training set and then the performance of this model will converge to the most frequent label in Table 4.1. This result shows the noisy nature of the image features in its raw form obtained from ResNet-50 Image Classification model as far as sense prediction is concerned. It highlights the need to transform the image features for our task.

The linear transformation of the Image Features by the Feed-Forward Neural Network model improves upon the performance of the k-Nearest Neighbours model and the Most Frequent Label model. This improvement, although small, shows that images do seem to have some information that could be used for identifying sense translations. It is important to note that besides the noisy nature of image features, a major reason for the poor performance of both, k-Nearest Neighbours and Feed-Forward Neural Network multi-class classifiers is the sheer number of classes. We have 3045 sense classes for English-German and 1970 sense

Language Pair	Model	Test 2016	Test 2017 Flickr	Test 2017 MSCOCO	Test 2018
English-German	k-Nearest Neighbours	1.8	0.2	2.4	3.4
	Feed-forward Neural Network	3.8	7.4	4.3	17.2
English-French	k-Nearest Neighbours	2.3	1.9	2.8	3.5
	Feed-Forward Neural Network	9.4	9.3	6.2	12.6
	Visually Ambiguous subsets of the test sets				
English-German	k-Nearest Neighbours	3.8	0.0	4.2	2.2
	Feed-forward Neural Network	3.8	8.2	9.2	21.4
English-French	k-Nearest Neighbours	0.4	1.8	3.9	2.0
	Feed-Forward Neural Network	3.3	6.4	9.3	8.8

Table 4.3: *k-Nearest Neighbours and Feed Forward Neural Network for Image-only Sense Label Prediction. These models assign a sense label to a given image. Image features taken from `pool5` layer of pre-trained ResNet-50 Image Classification model. Performance measured in Lexical Translation Accuracy (see Equation 3.9) as percentage between 0 and 100.*

classes for English-French. This also highlights the importance of specifying the ambiguous words as an input which reduces the sense classes from thousands to fewer than 10.

Next, we consider the Image-only Word Sense Translation task where we need to model $p(y|x, v)$. Again, we use k-Nearest Neighbours and Feed-Forward Neural Network. In k-Nearest Neighbours, we use the ambiguous word x to get all its sense-translations Y_x which is a subset of all sense-translation labels Y. Then for a new feature f, we simply retrieve the k nearest features which have a label in Y_x (and not Y). In other words, we are performing k-Nearest Neighbour search for each ambiguous word separately. We found the best results for $k = 10$ and Manhattan Distance. In Feed-Forward Neural Network, we concatenated the word embedding of the ambiguous word x and the image feature v to get a multimodal embedding. This multimodal embedding is fed into the Feed-Forward Neural Network as the input feature. These models were tested only on the Test 2016 dataset. The results are shown in Table 4.4.

We find that the k-Nearest Neighbour model is almost identical to the Most Frequent Translation model. The k = 30 nearest features obtained mostly have the most frequent translation as their label. This is expected because of the noisy nature of the image features in its raw form obtained from ResNet-50 Image Classification model as far as Word Sense Translation is concerned. It does not seem to provide more information to the ambiguous word. However, we do see improvements made by the Feed-Forward Neural Network over the Most Frequent Translation baseline signifying that the image features, if transformed, may hold some useful contextual information to correctly translate the ambiguous word.

Next, we consider the task of Text-only Sense Label Prediction $p(y|t)$. In this case, the textual context t has the ambiguous word x in it so we expected our models in this variant of our task to perform at par with the Most Frequent Translation model in Table 4.2. Again, we used k-Nearest Neighbours and Feed-forward Neural Network. The input feature f corresponding to the text t was the bag-of-words with Term Frequency - Inverse Document Frequency (TF-IDF) scores (see Appendix A.9). We first processed the input text t in the training set using the pre-processing steps in section 3.1.1 and then removed stop words and built a vocabulary V of the remaining words which was 6023 in size denoted as $V =$

Language Pair	Model	Test 2016	Difference w.r.t MFT
	Most Frequent Translation (MFT)	65.3	0
English-German	k-Nearest Neighbours	65.5	0.2
	Feed-Forward Neural Network	67.6	2.3
	Most Frequent Translation (MFT)	77.7	0
English-French	k-Nearest Neighbours	77.7	0
	Feed-Forward Neural Network	78.2	0.5
	Visually Ambiguous subsets of the test sets		
	Most Frequent Translation (MFT)	65.6	0
English-German	k-Nearest Neighbours	66.8	1.2
	Feed-Forward Neural Network	67.1	1.5
	Most Frequent Translation (MFT)	77.1	0
English-French	k-Nearest Neighbours	77.1	0
	Feed-Forward Neural Network	78.0	0.9

Table 4.4: *k-Nearest Neighbours and Feed Forward Neural Network for Image-only Word Sense Translation. The ambiguous word is specified in this task and the models are used to model $p(y|x, v)$. In k-Nearest Neighbours, we search for only those k nearest features in the training set which have a label which is a sense-translation candidate of the specified ambiguous word. In Feed-Forward Neural Network, Image features taken from `pool5` layer of pre-trained ResNet-50 Image Classification model are concatenated with word embedding of the specified ambiguous word. This way we get a multimodal embedding which is then fed into the Feed-Forward Neural Network. Performance is measured in Lexical Translation Accuracy (see Equation 3.9) as percentage between 0 and 100.*

$\{w_1, w_2, ..., w_{6023}\}$. We then created a 6023 dimensional feature vector $f = (f_1, f_2, ..., f_{6023})$ of every input sentence t where f_i is the TF-IDF score (see Appendix A.9) of the word w_i if $w_i \in t$, or else $f_i = 0$. This bag-of-words with TF-IDF scores features is extracted for each sentence and given as input to our models. The k-Nearest Neighbour model performed the best when k = 30 for Manhattan distance. The results of our models are presented in Table 4.5 below.

Surprisingly, the performance of both, k-Nearest Neighbours and Feed-Forward Neural Network models is much lower than the Most Frequent Translation model. There are three reasons for this: (1) as mentioned earlier, the number of sense classes is high, (2) input feature of 6023 dimension is extremely sparse with most of the entries being zeroes and (3) a sentence t can have multiple ambiguous words. We have, on average, 1.6 ambiguous words per sentence for English-German and 1.3 ambiguous words per sentence for English-French (see Table 3.1). Many sentences have multiple ambiguous words and as a result our model gets confused as for which ambiguous word it has to predict the sense-translation. This is an important point which made us realise that we need to reformulate our task from multi-class classification into a sequence tagging task.

Language Pair	Model	Test 2016	Test 2017 Flickr	Test 2017 MSCOCO	Test 2018
English-German	k-Nearest Neighbours	15.2	12.7	7.9	18.7
	Feed-forward Neural Network	36.8	30.4	25.3	39.2
English-French	k-Nearest Neighbours	23.0	18.7	14.0	23.4
	Feed-Forward Neural Network	42.4	37.3	34.6	40.6
Visually Ambiguous subsets of the test sets					
English-German	k-Nearest Neighbours	15.5	13.5	7.7	22.1
	Feed-forward Neural Network	37.7	31.2	26.0	40.8
English-French	k-Nearest Neighbours	23.5	19.1	14.3	25.4
	Feed-Forward Neural Network	42.1	37.3	34.5	39.4

Table 4.5: *k-Nearest Neighbours and Feed Forward Neural Network for Text-only Sense Label Prediction. These models assign a sense label to a given sentence. Bag-of-words features with TF-IDF scores are extracted from each input sentence and then fed to the model. Performance measured in Lexical Translation Accuracy (see Equation 3.9) as percentage between 0 and 100.*

4.3 Sequence Tagging: Sense Translation of Each Word in the Sentence

In this version of the task, the objective is to tag/label every word in the input sentence to it's correct sense-translation. For unambiguous words in our sentence, we have two options: (a) Tag these to a common tag, say an underscore '_' or (b) tag these words to themselves (see Figure 4.2). We call the first approach *ambiguous words only* and the latter approach *all words*.

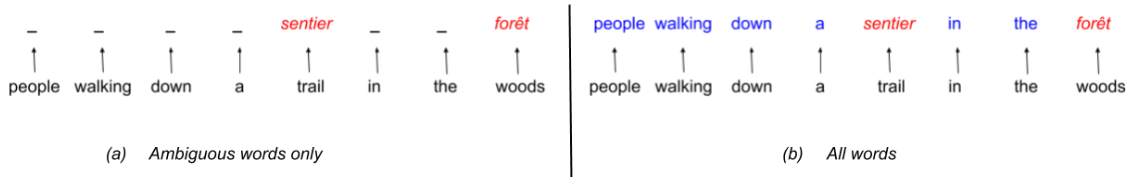


Figure 4.2: *Example of tagging every word in a sentence to their correct sense translations. In (a) unambiguous words are tagged to an underscore '_'. In (b) unambiguous words are tagged to themselves.*

Notice that in such a tagging task, we are not specifying the ambiguous word(s) in the input sentence, so we may still consider it to be a Text-only Sense Label Prediction task $p(y|t)$. However, even if we did specify the ambiguous word by a different token, it will still be the same. In other words, in the sequence tagging variant, Sense Label Prediction $p(y|t)$ and Word Sense Translation $p(y|x,t)$ are the same. We transformed our dataset to be in the format as shown in Figure 4.2 and developed tagging models for Monomodal and Multimodal Word Sense Translation. Another data-setting we explored is to include sentences from Multi30K which do not have any ambiguous words. Thus we have two more versions of training set: (a) *Ambiguous sentences only* which are sentences consisting at least one ambiguous word and (b) *All sentences* which has Ambiguous sentences plus other sentences with no ambiguous words. The *All sentences* training set is larger than *Ambiguous sentences only* training set by 16% for English-German and 21% for English-French.

4.3.1 Sequence Tagging using Long Short-Term Memory Network

Our models for Multimodal Word Sense Translation are based on Long Short-Term Memory Network so we will now take a closer look at it's internal architecture as depicted in Figure 4.3. A Long Short-Term Memory Network can be thought of as a blackbox which transforms or processes an input signal into a useful output signal. It is used recurrently, i.e. its output from the previous time-step is fed as one of it's inputs in the current time step. It has two internal states which are referred to as (a) Cell State or Memory denoted by c and (b) Hidden State or Activation denoted by h. Given an input sequence of vectors $x = (x_1, x_2, ..., x_t, ..., x_{|x|})$, we will now look at how an input vector x_t at time-step t gets transformed. The memory vector c_{t-1} and activation vector h_{t-1} from the previous time-step $t-1$ along with input vector x_t of the current time-step t are fed as inputs to the Long Short-Term Memory network. The input vector x_t and the activation vector h_t undergo four different affine transformations called input gate i_t, output gate o_t, forget gate f_t and proposed Cell State $\tilde{c}_t$ (see Equation 4.4). Affine transformation refers to a linear transformation followed by an element-wise

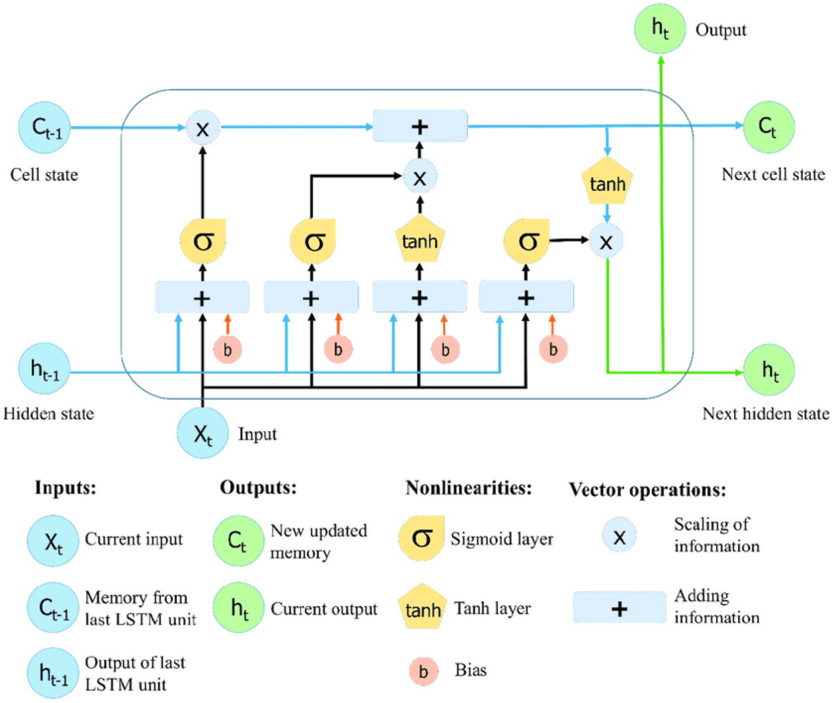


Figure 4.3: *The Flow of Information within Long Short-Term Memory Network.*

(non-linear) activation function (see Appendix A.4).

$$\begin{aligned} i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\ f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\ \tilde{c}_t &= \tanh(W_{\tilde{c}} x_t + U_{\tilde{c}} h_{t-1} + b_f) \end{aligned} \tag{4.4}$$

Out of the four affine transformations, the three gates are used for scaling of information; more specifically the three gates scale the different Cell States (Memory States) of the network c_{t-1}, $\tilde{c}_t$ and c_t. Finally, the previous Cell State c_{t-1} and the proposed Cell State $\tilde{c}_t$ are added after scaling to get the current or new Cell State c_t of the network. The new Cell State c_t undergoes further activation and scaling to get the current or new Hidden State h_t of the network as follows:

$$\begin{aligned} c_t &= (f_t \times c_{t-1}) + (i_t \times \tilde{c}_t) \\ h_t &= o_t \times \tanh(c_t) \end{aligned} \tag{4.5}$$

where × is element-wise multiplication.

Finally, the new Hidden State h_t and the new Memory State c_t are fed back to the network as inputs for the next time-step $t + 1$. The flow of information, described in the above Equations 4.4 and 4.5, within the Long Short-Term Memory network is depicted in Figure 4.3. The Hidden State h_t can be used as an output feature which can be fed to a Feed-Forward Neural Network, like in Figure 4.1. We call this the Softmax layer because the Softmax function (see Appendix A.12) is used to predict a class. One can think of h_t as modelling $p(y_t|x_1, x_2, ..., x_t)$ where y_t is the tag of x_t. Please note, only partial context (x_1 to x_t) is being considered and not the full context (x_1 to $x_{|x|}$). We can use a Long Short-Term Memory network for our task of tagging each word in the input sentence to its correct sense-preserving lexical translation as depicted below in Figure 4.4.

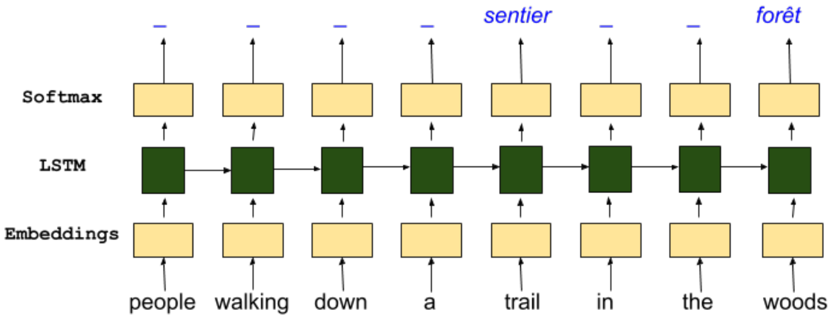


Figure 4.4: *Long Short-Term Memory (LSTM) Network for Tagging Word Sense Translations. We have shown for the ambiguous words only configuration (see Figure 4.2). Embeddings layer is a look-up table that embeds a word into a corresponding vector. Softmax layer is a Feed-Forward Neural Network which uses the Hidden State of the Long Short-Term Memory Network as an input to predict a tag using the Softmax function (see Appendix A.12).*

In the above Figure 4.4, when tagging the ambiguous word *trail*, only partial context is being used by the Long Short-Term Memory network; i.e. only words to the left of *trail* are being considered. So, we also developed a Bi-directional Long Short-Term Memory network. This uses two separate Long Short-Term Memory networks, one that reads words in a sentence from left to right and the other which reads the sentence from right to left. Then the Hidden States corresponding to a particular word in both the networks is concatenated and fed into the Softmax layer. This is depicted in the Figure 4.5. The concatenated Hidden States can be thought of as modelling $p(y_t|x_1, x_2, ..., x_t)$ where y_t is the tag of x_t at time-step t. It uses the entire context (all words in the input sentence). Bidirectional Long Short-Term Memory networks are also known to be the state-of-the-art systems on other tagging tasks like multilingual Part-of-Speech tagging (Plank et al. 2016). We also explored having two layers of Long Short-Term Memory networks stacked one over the other for both Unidirectional and Bidirectional Long Short-Term Memory networks. Also, we explored Encoder-Decoder Neural Machine Translation architecture using two Long Short-Term Memory networks identical to Figure 2.2 in Chapter 2.

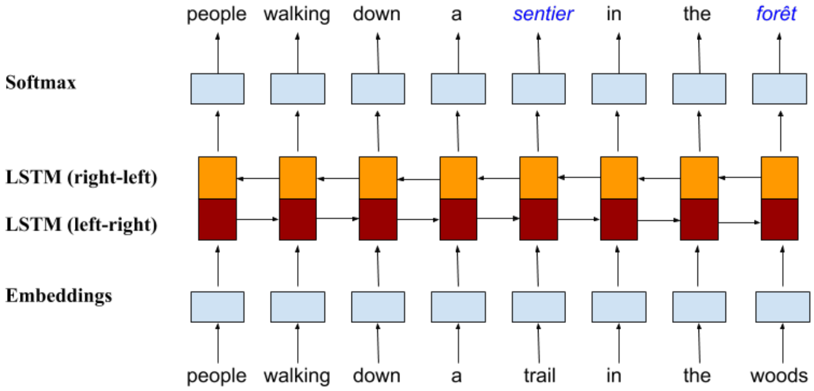


Figure 4.5: *Bidirectional Long Short-Term Memory Network for Tagging Sense Translations. We have shown for the all words configuration (see Figure 4.2). Embeddings layer is a look-up table that embeds a word into a corresponding vector. Softmax layer is a Feed-Forward Neural Network which uses the concatenated Hidden States of the two Long Short-Term Memory Networks as an input to predict a tag using the Softmax function (see Appendix A.12).*

In our experiments with these architectures, the dimension of the Cell State c and the Hidden State h is set to 300. We also use 300 dimensional word embeddings. As a result, a single Long Short-Term Memory network has 721,200 parameters which is much larger than our dataset size. So it is important to note that our models were learning in an over-parameterised regime. The model parameters of all our models were learnt by training them on the training set with different data-settings using backpropagation algorithm and cross-entropy loss function (see Appendix A.14). For optimization, we used the Adam (Kingma & Ba 2014) algorithm with different learning rates between 0.0001 and 0.01 and batch size was chosen from {16, 32, 64}. Other hyperparameters include dropout, which was chosen from {0.2, 0.3, 0.4, 0.5}, and early stopping where training is stopped if model accuracy over the validation set does not improve for 30 epochs. Then the best performing model over the validation set was selected. All our models were implemented and trained in the TensorFlow (Abadi et al. 2016) framework.

4.3.2 Results and Discussion

The performance of our sequence tagging models for text-only Word Sense Translation in different data-settings are shown in Table 4.6 for English-German and in Table 4.7 for English-French. We will first analyse the English-German results. We observe the Bidirectional Long Short-Term Memory network outperforms every other model in all data-settings for Test 2016 and Test 2017 Flickr datasets. In the Test 2017 MSCOCO dataset, it outperforms other models except in *ambiguous sentences + ambiguous words* data-setting and *all sentences + all words* data-setting. This can be attributed to the fact that the Test 2016, Test 2017 Flickr and the Training dataset come from the same distribution (all three are Flickr datasets), while Test 2017 MSCOCO dataset has a different distribution than the Training set. The second surprising observation is that the deeper models consisting 2 stacked layers of Unidirectional and Bidirectional Long Short-Term Memory networks perform poorly. We suspect this is because of the Vanishing Gradient problem (Hochreiter 1991) where error does not sufficiently backpropagate to the first layer (Gradient diminishes) because of which it is under-trained. We could resolve this with residual connections like in ResNet (He et al.

Model	Test 2016	Test 2017 Flickr	Test 2017 MSCOCO
Most Frequent Translation	65.3	60.5	52.5
all sentences + ambiguous words			
Long Short-Term Memory	64.0	55.6	46.7
2 layers of Long Short-Term Memory	57.9	50.5	40.4
Bidirectional Long Short-Term Memory	**67.6**	**61.7**	**54.3**
2 layers of Bidirectional Long Short-Term Memory	60.5	53.4	43.3
Encoder-Decoder using Long Short-Term Memory	63.8	54.5	50.1
ambiguous sentences + ambiguous words			
Long Short-Term Memory	63.6	57.4	49.1
2 layers of Long Short-Term Memory	55.8	50.1	36.7
Bidirectional Long Short-Term Memory	**68.2**	**62.1**	52.8
2 layers of Bidirectional Long Short-Term Memory	60.0	51.8	40.2
Encoder-Decoder using Long Short-Term Memory	64.2	53.5	**55.6**
all sentences + all words			
Long Short-Term Memory	66.6	59.3	**54.1**
2 layers of Long Short-Term Memory	58.3	50.8	40.7
Bidirectional Long Short-Term Memory	**69.0**	**60.5**	**54.1**
2 layers of Bidirectional Long Short-Term Memory	61.0	51.8	44.1
Encoder-Decoder using Long Short-Term Memory	65.2	57.6	**54.1**
ambiguous sentences + all words			
Long Short-Term Memory	67.3	59.9	57.0
2 layers of Long Short-Term Memory	60.4	51.2	43.9
Bidirectional Long Short-Term Memory	**69.6**	**62.4**	**57.2**
2 layers of Bidirectional Long Short-Term Memory	62.4	52.2	42.3
Encoder-Decoder using Long Short-Term Memory	66.5	55.6	51.7

Table 4.6: *Text-only Word Sense Translation using Sequence models in different data settings for English-German. Performance measured in Lexical Translation Accuracy (see Equation 3.9) as percentage between 0 and 100. The best performing models and data-settings are in* ***Bold Font****.*

2016) to bypass a layer. However, we believe our task is much simpler and does not need many stacked layers of Long Short-Term Memory networks because it will further increase the number of parameters of the already over-parameterized model. An interesting observation is that Neural Machine Translation-style Encoder-Decoder model performs at par with the Unidirectional Long Short-Term Memory network. In some test sets and data-settings, one performs better than the other and vice-versa in other test-sets and data-settings. The two models have different advantages. On one hand, in Unidirectional Long Short-Term Memory network (see Figure 4.4), a word x_t at time-step t gets tagged right away at the same time-step while in Encoder-Decoder (see Figure 2.2) the Encoder time-step t and the Decoder time-step t are l distance apart where l is the length of the sentence. On the other hand, an Encoder-Decoder model tags a word with it's sense translation after having read the entire sentence while in Unidirectional Long Short-Term Memory network only partial context is considered as mentioned in the earlier subsection 4.3.1.

Another surprising result is that the models trained on *all words* data-setting tend to outperform the same models trained on *ambiguous words* data-setting. This is surprising because in *all words* data-setting, the number of classes/tags is much larger than the number of classes/tags in *ambiguous words* data-setting (unambiguous words are tagged to themselves in *all words* data-setting). In *all words* data-setting we have more than 13 thousand unique tags while in *ambiguous words* data-setting we have a little more than 3 thousand unique tags. We hypothesize that tagging the unambiguous words to themselves forces the model to capture the context better. Also, we observe that the models tend to perform slightly better in the *ambiguous sentences* data-setting as compared to the *all sentences* data-setting. This hints that more data is not always better, as unambiguous sentences, which don't have ambiguous words, are not always relevant to the task. This is in line with the observations in (Postma et al. 2016). Next, we analyse the English-French results and observe similar findings except that Encoder-Decoder Neural Machine Translation model architecture is slightly worse than Long Short-Term Memory network. We highlight that no model was able to beat the Most Frequent Translation baseline model for the Test 2017 Flickr dataset. Finally, we conclude Bidirectional Long Short-Term Memory network trained on the *ambiguous sentences* and *all words* data-setting is the best performing model for our task. Next, we will explore conditioning our sequence tagging models on the image features.

4.4 Sequence Tagging for Multimodal Word Sense Translation

For Multimodal Word Sense Translation, we need to model $p(y|x, t, v)$. Our sequence tagging systems so far model $p(y|x, t)$. To incorporate vision modality v, we take inspiration from Multimodal Machine Translation models in Section 2.4.2. More specifically, we begin by using image features to initialise the Long Short-Term Memory units by setting the Hidden State h_0 and Cell State c_0 at time-step 0 to the image features. This is depicted below in Figure 4.6. Please note, there is a mismatch in the dimensions of the Pooled Image Features and the dimensions of the Hidden State and Cell States of the Long Short-Term Memory Network. Image features from the `pool5` layer of ResNet-50 is 2048 dimensional while Hidden State h and Cell State c are 300 dimensional. So, we introduced an affine transformation of the Image Features to reduce its dimensions to 300. The weights of this affine transformation are also learnt during training. We experimented with initialising the Unidirectional Long

Model	Test 2016	Test 2017 Flickr	Test 2017 MSCOCO
Most Frequent Translation	77.7	**77.3**	67.1
	all sentences + ambiguous words		
Long Short-Term Memory	73.7	72.0	**67.3**
2 layers of Long Short-Term Memory	66.7	73.1	59.7
Bidirectional Long Short-Term Memory	**76.9**	**74.6**	65.5
2 layers of Bidirectional Long Short-Term Memory	69.7	64.7	56.9
Encoder-Decoder using Long Short-Term Memory	71.3	68.1	62.8
	ambiguous sentences + ambiguous words		
Long Short-Term Memory	74.4	72.7	**67.3**
2 layers of Long Short-Term Memory	66.6	65.1	55.3
Bidirectional Long Short-Term Memory	**78.6**	**75.2**	**67.3**
2 layers of Bidirectional Long Short-Term Memory	69.8	66.6	58.5
Encoder-Decoder using Long Short-Term Memory	75.9	70.3	64.6
	all sentences + all words		
Long Short-Term Memory	76.5	**73.9**	68.0
2 layers of Long Short-Term Memory	69.2	67.0	60.5
Bidirectional Long Short-Term Memory	**78.4**	73.8	**68.7**
2 layers of Bidirectional Long Short-Term Memory	70.5	66.9	62.6
Encoder-Decoder using Long Short-Term Memory	73.8	71.6	64.6
	ambiguous sentences + all words		
Long Short-Term Memory	78.2	76.4	**71.2**
2 layers of Long Short-Term Memory	70.1	57.7	61.4
Bidirectional Long Short-Term Memory	**80.4**	**76.8**	70.5
2 layers of Bidirectional Long Short-Term Memory	72.2	67.6	63.3
Encoder-Decoder using Long Short-Term Memory	76.4	72.9	69.4

Table 4.7: *Text-only Word Sense Translation using Sequence models in different data settings for English-French. Performance measured in Lexical Translation Accuracy (see Equation 3.9) as percentage between 0 and 100. The best performing models and data-settings are in* ***Bold Font****.*

Short-Term Memory Network and Bidirectional Long Short-Term Memory Network with the Image Features. The code base for our models is made available at `https://github.com/ImperialNLP/mltcode`. The results of our experiments are presented in Table 4.8 for English-German and in Table 4.9 for English-French.

4.4.1 Results and Discussions

We will first analyze the English-German results. It can be clearly seen that Unidirectional Long Short-Term Memory network benefits from the initialization of its Hidden States with the Image Features. In 11 out of 12 test scores, we see that the image initialized network gains over its text-only version. However, for Bidirectional Long Short-Term Memory Network, the impact of initializing the hidden and memory states with Image Features is inconclusive. In 6 out of 12 test scores, the image initialized network gains over its text-only version and in the remaining 6 test scores the performance drops with the image features. We believe the reason is that the Unidirectional Long Short-Term Memory Network models the probability of a tag

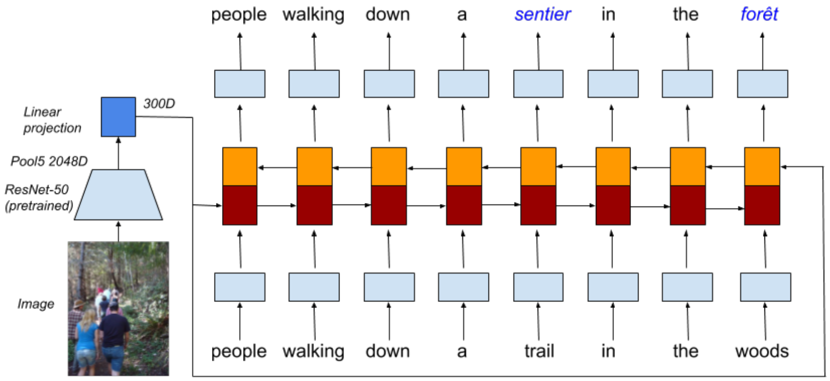


Figure 4.6: *Bidirectional Long Short-Term Memory Network Initialised with Image Features for Multimodal Word Sense Translations. We have shown for the all words configuration in this figure. For all words configuration, please refer Figure 4.2. Image features undergo an affine transformation to match their dimensions to the dimensions of the Hidden state and Cell State of the Long Short-Term Memory networks.*

y_t at time-step t conditioned only on the partial textual context; i.e. $p(y_t|v, x_1, x_2, ..., x_t)$ and not on the full textual context $p(y_t|v, x_1, x_2, ..., x_{|x|})$. The model has seen only those words which are to the left of the current word. On the other hand, a Bidirectional Long Short-Term Memory network reads the entire textual context when tagging a word to its sense translation. Initializing the Hidden State and the Memory State of the network with the image features / visual context seems to compensate for the lack of textual context in the Unidirectional Long Short-Term Memory network and that's why it benefits from the image. This has also been observed in Caglayan et al. (2019) where Multimodal Machine Translation system benefits from image features when the textual context is reduced.

Another interesting observation is that the gain from the image features is more in the *ambiguous words* data-setting compared to *all words* data-setting. Recall from Figure 4.2, in the *ambiguous words* data-setting, the model is tagging all ambiguous words to an underscore '_' and in the *all words* data-setting, the model is tagging unambiguous words to themselves. We believe, tagging unambiguous words to themselves forces the model to capture the textual context better. In other words, tagging the unambiguous words to a common token underscore '_' results in an inefficient representation of the textual context in the Hidden State and the Memory State of the Long Short-Term Memory units. Initialising these with the Image Features seems to compensate for the lack of textual context within the internal representations of the network. Again, we see the same theme where visual context benefits a model when textual context is either partial or inefficiently captured by our tagging models.

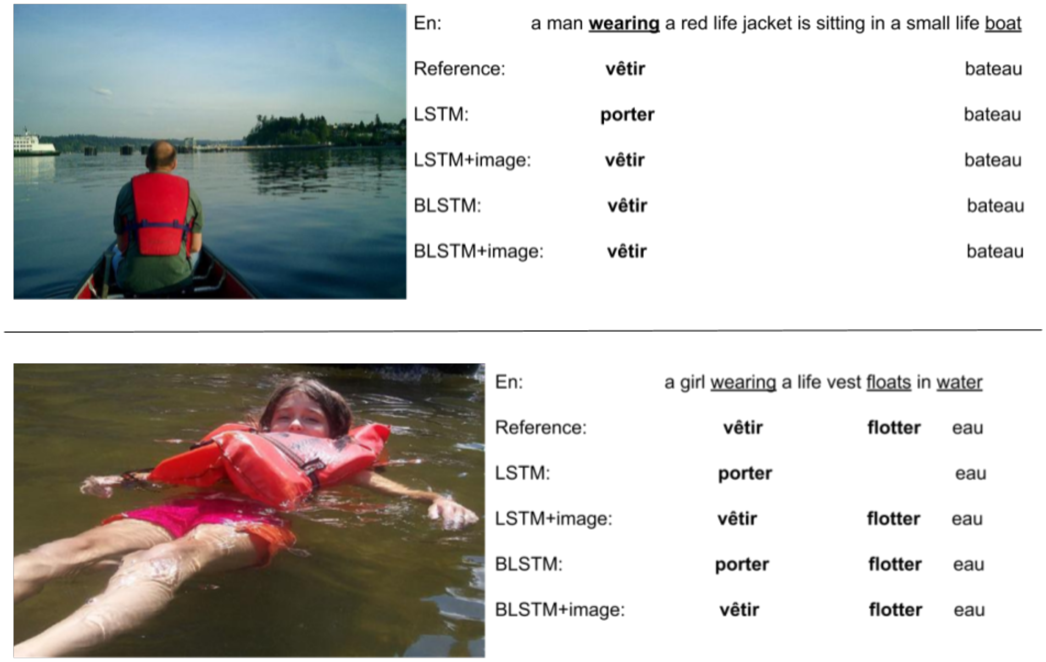
The English-French results shown in Table 4.9 above are similar to the English-German results. The unidirectional Long Short-Term Memory network benefits from being initialised with the Image Features in 10 out of 12 Test scores. The Bidirectional Long Short-Term Memory network gains from being initialised with the Image Features in 6 out 12 Test results. Also, the improvements from the image tend to be more in the *ambgiuous words* data-setting.

A couple of examples which show that unidirectional Long Short-Term Memory network benefits more from the `pool5` Image Features from ResNet-50 as compared to a Bidirectional

Model	Test 2016	Test 2017 Flickr	Test 2017 MSCOCO
Most Frequent Translation	65.3	60.5	52.5
all sentences + ambiguous words			
LSTM	64.0	55.6	46.7
LSTM + Image	**66.1 (+2.1)**	**58.4 (+2.8)**	**52.2 (+5.5)**
BLSTM	67.6	61.7	54.3
BLSTM + Image	**68.4 (+0.8)**	60.9 (-0.8)	53.8 (-0.5)
ambiguous sentences + ambiguous words			
LSTM	63.6	57.4	49.1
LSTM + Image	**66.3 (+2.7)**	**59.2 (+1.8)**	**54.3 (+5.2)**
BLSTM	68.2	62.1	52.8
BLSTM + Image	**68.6 (+0.4)**	60.1 (-2.0)	**54.1 (+1.3)**
all sentences + all words			
LSTM	66.6	59.3	54.1
LSTM + Image	**66.9 (+0.3)**	**59.7 (+0.4)**	**54.9 (+0.8)**
BLSTM	69.0	60.5	54.1
BLSTM + Image	68.7 (-0.3)	59.7 (-0.8)	**55.4 (+1.3)**
ambiguous sentences + all words			
LSTM	67.3	59.9	57.0
LSTM + Image	**67.6 (+0.3)**	**60.0 (+0.1)**	55.6 (-1.4)
BLSTM	69.6	62.4	57.2
BLSTM + Image	**69.8 (+0.2)**	60.8 (-1.6)	**57.5 (+0.3)**
Best Performing model on the Visually Ambiguous subset			
BLSTM	69.9	63.7	59.8
BLSTM + Image	**70.5 (+0.6)**	63.3 (-0.4)	59.8 (+0.0)

Table 4.8: *Multimodal Word Sense Translation using Sequence models in different data settings for English-German. The Hidden State and the Cell State of Long Short-Term Memory units are initialised with the Image Features as shown in Figure 4.6. LSTM stands for Long Short-Term Memory. BLSTM stands for Bidirectional LSTM. Performance is measured in Lexical Translation Accuracy (see Equation 3.9) as percentage between 0 and 100. If Multimodal models improve upon their text-only baselines then their scores are highlighted in **Bold Font**.*

Long Short-Term Memory network are shown below in Figure 4.7. We see three interesting examples of Multimodal Word Sense Translation. Consider the ambiguous word ‘wearing’ in the first example. It is the third word of the sentence. When tagging this word to its sense translation the unidirectional network has only seen the two words to its left “a man” as its textual context. With a partial textual context, the unidirectional network tags ‘wearing’ to ‘porter’ (carry) in French. The bidirectional network which has seen the entire sentence tags ‘wearing’ correctly to ‘vêtir’ (wear) in French. The unidirectional network which is initialized with the Image Feature, correctly translates ‘wearing’ to ‘vêtir’ which shows that the Visual



En: a man **wearing** a red life jacket is sitting in a small life boat

Reference:	**vêtir**	bateau
LSTM:	**porter**	bateau
LSTM+image:	**vêtir**	bateau
BLSTM:	**vêtir**	bateau
BLSTM+image:	**vêtir**	bateau

En: a girl wearing a life vest floats in water

Reference:	**vêtir**	**flotter**	eau
LSTM:	**porter**		eau
LSTM+image:	**vêtir**	**flotter**	eau
BLSTM:	**porter**	**flotter**	eau
BLSTM+image:	**vêtir**	**flotter**	eau

Figure 4.7: *Long Short-Term Memory Networks benefit from Initialization with Image Features. LSTM stands for Long Short-Term Memory network. BLSTM stands for Bidirectional LSTM. '+image' stands for initializing the Hidden State and the Memory State with the Image Features. In the first example, the ambiguous word* wearing *is incorrectly translated to* porter *in French by the LSTM. However, when the Hidden State and Memory State of the LSTM are initialised with the Image features then the model translated the correct sense of the word. Notice the word* wearing *appears early in the sentence, so the LSTM has only seen the two words to its left "a man" and not the entire sentence when tagging the word* wearing*. This is an example of partial textual context benefiting from the visual context.*

Model	Test 2016	Test 2017 Flickr	Test 2017 MSCOCO
Most Frequent Translation	77.7	77.3	67.1
all sentences + ambiguous words			
LSTM	73.7	72.0	67.3
LSTM + Image	**75.6 (+1.9)**	**72.7 (+0.7)**	66.7 (-0.6)
BLSTM	76.9	74.6	65.5
BLSTM + Image	**77.7 (+0.8)**	72.9 (-1.7)	65.3 (-0.2)
ambiguous sentences + ambiguous words			
LSTM	74.4	72.7	67.3
LSTM + Image	**76.9 (+2.5)**	**73.8 (+1.1)**	**68.0 (+0.7)**
BLSTM	78.6	75.2	67.3
BLSTM + Image	**79.1 (+0.5)**	74.2 (-1.0)	66.9 (-0.4)
all sentences + all words			
LSTM	76.5	73.9	68.0
LSTM + Image	**77.1 (+0.6)**	**74.4 (+0.5)**	**69.2 (+1.2)**
BLSTM	78.4	73.8	68.7
BLSTM + Image	**79.9 (+1.5)**	**73.9 (+0.1)**	**70.7 (+2.0)**
ambiguous sentences + all words			
LSTM	78.2	76.4	71.2
LSTM + Image	**78.3 (+0.1)**	**76.5 (+0.1)**	70.5 (-0.7)
BLSTM	80.4	76.8	70.5
BLSTM + Image	80.4 (0.0)	75.9 (-0.9)	**70.7 (+0.2)**
Best Performing model on the Visually Ambiguous subset			
BLSTM	80.0	77.2	71.3
BLSTM + Image	79.7 (-0.3)	77.0 (-0.2)	**71.8 (+0.5)**

Table 4.9: *Multimodal Word Sense Translation using Sequence models in different data settings for English-French. The Hidden State and the Cell State of Long Short-Term Memory units are initialised with the Image Features as shown in Figure 4.6. LSTM stands for Long Short-Term Memory. BLSTM stands for Bidirectional LSTM. Performance is measured in Lexical Translation Accuracy (see Equation 3.9) as percentage between 0 and 100. If Multimodal models improve upon their text-only baselines then their scores are highlighted in **Bold Font**.*

Context is compensating for the lack of textual context. In the second example, both the text-only models, unidirectional and bidirectional networks, translate 'wearing' incorrectly and these benefit from being initialised with the Image Features. Finally, the unidirectional network does not consider 'floats' as an ambiguous word but when initialized with Image Features, it translates 'floats' correctly to 'flotter' in French.

Other ways of incorporating the vision modality into our sequence tagging models were also considered. Inspired by Multimodal Machine Translation strategies in Section 2.4.2, we tried (a) concatenating Image Feature to Word Embeddings and (b) using Image Features as separate words. We conducted quick experiments to see if we got anything different from our approach of initialising Hidden States with image features in Tables 4.8 and 4.9. The findings were similar so we shifted our focus on using our Multimodal Word Sense Translation models for other downstream tasks.

4.5 Multimodal Word Sense Translation to re-rank Machine Translation outputs

Word Sense Disambiguation was originally conceived as a task that can benefit Machine Translation. Similarly, our intention to develop Multimodal Word Sense Translation models was to later use it in Machine Translation and improve the quality of translations. Modern Neural Network approaches of Machine Translation are trained end-to-end. So, utilising our Multimodal Word Sense Translation models into a Neural Machine Translation system is a challenge. We explored a pipeline approach of re-ranking 'n-best translation candidates' of a Neural Machine Translation system using our models.

Machine Translation systems, both Statistical Machine Translation and Neural Machine Translation, can be made to generate a list of 'n-best' translation candidates for a given source sentence. This is done using the 'beam search' strategy which we formally describe as follows: In Neural Machine Translation, a decoder generates a translation word-by-word from left to right that maximizes the conditional probability $p(y_1^t|x)$ where y_1^t refers to sequence of t words $(y_1, y_2, ..., y_t)$. At time step t it uses a probability distribution $p(y_t|x, \hat{y}_1^{t-1})$ to get the most probable word $\hat{y}_t$ at that time-step conditioned on the source sentence x and the sentence generated so far $\hat{y}_1^{t-1}$. In other words,

$$\hat{y}_t = \underset{y_t \in Y}{\arg\max}\, p(y_t|x, \hat{y}_1^{t-1}) \tag{4.6}$$

where Y is the Vocabulary of the Target Language.

However, instead of generating just one word $\hat{y}_t$ at time-step t in the above Equation 4.6, the decoder can also be made to generate k most probable words from the probability distribution $p(y_t|x, \hat{y}_1^{t-1})$. We refer to k as the 'beam' size. So in the first time-step $t = 1$, we get k words. Then in the next time-step $t = 2$, for every word we had generated, we get k more following words. Hence, we have k^2 translation candidates so far. We prune these k^2 candidates to the n most probable candidates. This process of (a) generating k words at each time-step for every translation candidate that has been generated so far followed by (b) pruning to n most probable translations is continued. Eventually, we get the n-best translations of the given source sentence ranked by their probability scores. Out of these, the translation with the highest probability score is returned by the system. More beam search strategies for Neural Machine Translation are discussed in Freitag & Al-Onaizan (2017). The beam search decoding in Statistical Machine Translation is similar to what we have described above where Dynamic Programming algorithms are used to obtain the n-best translations (Tillmann & Ney 2003). The beam search in phrase-based Statistical Machine Translation is slightly different because instead of individual words we have to deal with phrases (Koehn 2004).

In Multimodal Machine Translation, Shah et al. (2016) explored re-ranking the n-best translations generated by a Statistical Machine Translation system using the Image Features from ResNet-50. In our own previous work, we explored re-ranking the n-best translations generated by a Neural Machine Translation system using an Image Captioning system and found some potentially positive results (Lala et al. 2017). So, we continued exploring re-ranking options further. We built our own standard Neural Machine Translation system, which is an attention-based Encoder-Decoder architecture, for English-German, English-French and English-Czech language directions and noticed that the translation hypotheses besides the 1-best output were also of high quality. We made our systems produce 20-best translations for each source sentence in English in the Validation set of Multi30K and selected the translation with the highest sentence-level METEOR score. We call these selected translations over the entire Validation set as 'the Oracle'. Next, we compared the Oracle to the 1-best translation output. In this experiment, we observed that the Oracle performs way better (11 to 13.5 METEOR points) than the 1-best output (see Table 4.10). This shows that re-ranking of the n-best translations has a scope of improving the quality of translation by upto 13.5 METEOR points for English-German, by upto 12 METEOR points for English-French and by upto 11 METEOR points for English-Czech.

Language-Pair	1-best	Best of 20-best	Scope or difference
		(Oracle)	(Oracle - 1-best)
English-German	48.36	61.85	**+13.49**
English-French	64.91	76.87	**+11.96**
English-Czech	33.87	44.71	**+10.84**

Table 4.10: *Oracle Experiment to Evaluate the Scope of Re-Ranking n-best Translations from a standard Neural Machine Translation system. In this exploratory experiment, we observe that re-ranking of the 20-best translation hypotheses generated by a standard attention based Encoder-Decoder Neural Machine Translation model has the potential of improving translation by upto 13.49 METEOR points for the three language pairs.*

For a re-ranking strategy, we were inspired by how humans use images to translate image descriptions. We believe humans look at the image usually to disambiguate ambiguous words in the source sentence especially in those instances where the text alone is not sufficient. For example, translating '*A **sportsperson** is playing football*' into French requires us to know whether the sportsperson is a male or a female and accordingly the translation is '*Un **sportif** joue au football*' (male) or '*Une **sportive** joue au football*' (female). This example is depicted in Figure 4.8. In this case, we believe a human translator has the two translations at the back of his/her mind ranked in some order. After looking at the image, the human translator may re-rank the translation hypotheses in his/her mind and select the correct translation of the source sentence with the correct sense-translation of the ambiguous word which is what we have tried to mimic in our approach.

More specifically, in our systems we adopted a two-step pipeline approach. In the first step, we used various Neural Machine Translation strategies to produce lists of 10-best translation hypotheses. In the second step, we re-ranked the 10-best translation hypotheses using our Multimodal Word Sense Translation model which is a Bidirectional Long Short-Term Memory network with Hidden State and Memory State initialised by the Image Features from `pool5` layer of ResNet-50. For control experiments, we also re-ranked the 10-best trans-

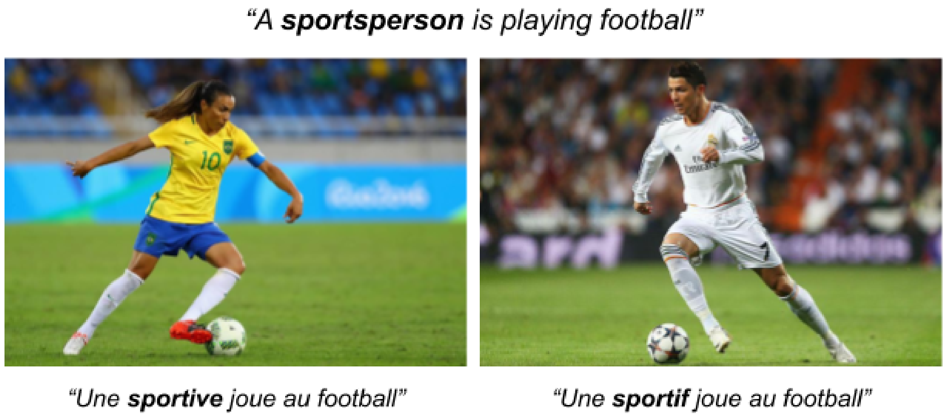

"A **sportsperson** is playing football"
"Une **sportive** joue au football"
"Une **sportif** joue au football"

Figure 4.8: *The Scope of using Images for Re-Ranking n-best Translations. A human would look at an image to correct the translation that has already been generated in his/her mind. In this example, for the given source sentence "A sportsperson is playing footbal", the two French Translations (a) "Un* ***sportif*** *joue au football" and (b) "Une* ***sportive*** *joue au football" are already in the n-best translations in the mind of a Human translator. Then the human looks at the image and decides to re-rank the n-best translations and chooses one translation over the other based on whether the sportsperson is a male or a female. In other words, human translator is using the visual context for disambiguating the ambiguous word 'sportsperson' and using it to re-rank the n-best translations in his/her mind.*

lations using the text-only Word Sense Translation model and the Most Frequent Translation model baselines.

For the first step of generating 10-best translation hypotheses, we made use of an ensemble of text only attention based Encoder-Decoder Neural Machine Translation models with Conditional Gated Recurrent Units (Cho et al. 2014) decoder. We built the systems using the NMTPY toolkit (Caglayan, García-Martínez, Bardet, Aransa, Bougares & Barrault 2017). Our models have a setting similar to Caglayan et al. (2016) with a Bidirectional Gated Recurrent Units with Hidden State of 256 dimensions as an Encoder followed by a Conditional Gated Recurrent Unit Decoder which is initialized with a non-linear transformation of the mean of the Encoder Hidden States. We used a simple Feed-Forward Neural Network to compute the attention scores as described in Equations 2.9, 2.10 and 2.11. We used Adam optimizer with a learning rate of 0.00005 and a batch size of 64. We set the Embedding dimensionality of the Encoder and the Decoder to 128 and followed the default parametrization in Caglayan, Aransa, Bardet, Garcia-Martinez, Bougares, Barrault, Masana, Herranz & van de Weijer (2017). Our final model is an ensemble of different runs of the basic Neural Machine Translation model with five different seeds. This system is then made to generate 10-best translation hypotheses for every source sentence using 'Beam search' described earlier. For more details on our Neural Machine Translation model, please refer to Lala et al. (2018).

Our re-ranking strategy is depicted in Figure 4.9. First, given an English source sentence, the Neural Machine Translation model generates an n-best list of translation candidates with a likelihood score given by the Decoder of the model. The idea is to select the translation

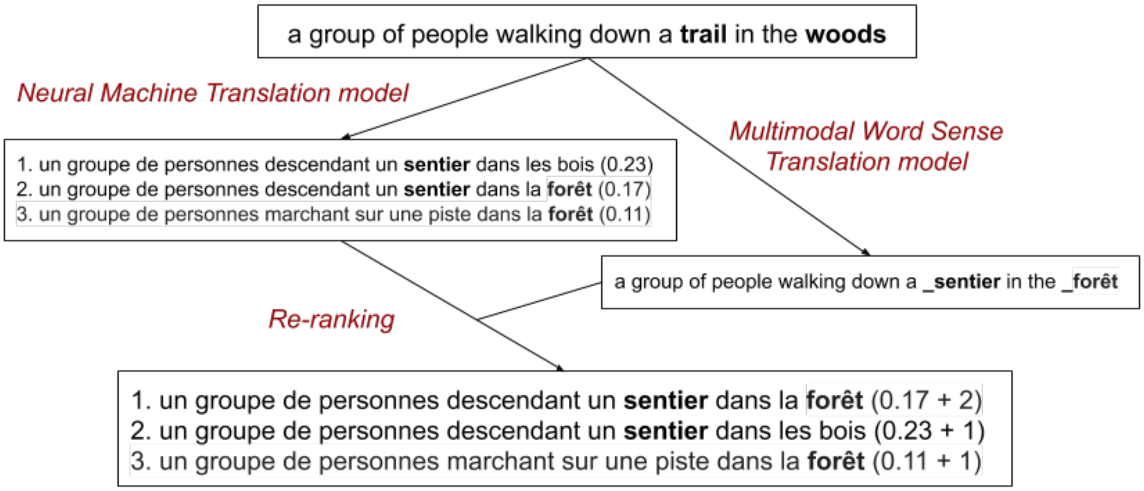

a group of people walking down a **trail** in the **woods**
Neural Machine Translation model
Multimodal Word Sense Translation model
1. un groupe de personnes descendant un **sentier** dans les bois (0.23)
2. un groupe de personnes descendant un **sentier** dans la **forêt** (0.17)
3. un groupe de personnes marchant sur une piste dans la **forêt** (0.11)
a group of people walking down a **_sentier** in the **_forêt**
Re-ranking
1. un groupe de personnes descendant un **sentier** dans la **forêt** (0.17 + 2)
2. un groupe de personnes descendant un **sentier** dans les bois (0.23 + 1)
3. un groupe de personnes marchant sur une piste dans la **forêt** (0.11 + 1)

Figure 4.9: *Re-Ranking n-best Translations Generated by a Neural Machine Translation system using Multimodal Word Sense Translation models. The Neural Machine Translation model generates n-best translation candidates of the source sentence. The Multimodal Word Sense Translation model translates ambiguous words in the source sentence to their correct sense-preserving lexical translations. The re-ranking step uses these lexical translations to re-score and re-rank the n-best translation candidates. Here, the re-scoring formula is to take the likelihood of the translation candidate and simply add the number of sense-preserving lexical translations found in both the translation candidate and in the output of Multimodal Word Sense Translation system. For example, we added 2 to the likelihood 0.17 of the translation candidate which contains both* sentier *and* forêt *found in the output of Multimodal Word Sense Translation model.*

candidate in the n-best translations which correctly disambiguates as many ambiguous words in the source sentence as possible. The source sentence in our example (Figure 4.9) contains two ambiguous words *trail* and *woods*. We used our best performing Multimodal Word Sense Translation model from the Table 4.9 to predict the sense translations of these words (the correct ones being *sentier* and *forêt* respectively in this example). Next, we match these to the words in the translation candidates and re-score the original likelihood scores by adding the number of matching words to it. Then, the n-best translations are re-ranked using the new scores and the top candidate (which has the highest number of matches) is used in the evaluation. Other 're-scoring' formulations, like a weighted average, were also experimented with but these didn't result in any change in the performance as measured using METEOR.

We used our re-ranking strategy in another task of Multimodal and Multilingual Machine Translation where we are given an image and its descriptions in English, German and French, and then we have to translate these into Czech. In this scenario, we built three text-only Neural Machine Translation systems, one for English-Czech, one for German-Czech and one for French-Czech just the way we described earlier. We then made each of the three systems generate 10-best translations in Czech resulting in total 30 translation candidates in Czech. We then used our re-ranking strategy on these 30 translations. Please note, we used only the English-Czech Multimodal Word Sense Translation model for re-ranking because we don't have a dataset of ambiguous words and its sense translations for German-Czech or French-

Czech on which we could train more Multimodal Word Sense Translation models for those language pairs.

We also developed other re-ranking/selection strategies like (a) Consensus-based selection where translation occurring in the 10-best lists of all three Neural Machine Translation systems of the three language pairs is selected, (b) Data augmentation followed by selection using Random Forest classifier, and, (c) Data augmentation followed by selection using Recurrent Neural Network classifier for the Multimodal and Multilingual Machine Translation task. For detailed description of these other strategies, please refer to Lala et al. (2018). These re-ranking/selection strategies are different from our re-ranking strategy using Multimodal Word Sense Translation. We have mentioned these other strategies just for comparison with our strategy.

4.5.1 Results and Discussions

We submitted all the above mentioned systems, with and without re-ranking, to the Multimodal Machine Translation shared task of 2018 (Barrault et al. 2018). These systems were evaluated using standard metrics for Machine Translation like BLEU, METEOR, TER and our metric of 'sense-preserving Lexical Translation Accuracy' (LTA). Human evaluation by Direct Assessment was also carried out for some systems. The results are presented in Table 4.11. There were many other systems submitted to the shared task but our focus is only on evaluating the impact of our re-ranking strategy; so, we have included only those systems that will help us understand the contribution of Multimodal Word Sense Translation to Neural Machine Translation via re-ranking of n-best translations.

First of all, we observe all our translation systems, with or without re-ranking, beat the baseline system provided by the task organizers. Next, for English-German, none of the re-ranking strategies change the METEOR score of the Neural Machine Translation system. On the other hand, re-ranking only worsens the quality of translation as measured using BLEU and TER metrics. On the positive side, our re-ranking strategy of using Multimodal Word Sense Translation system improves LTA score which shows that our approach is helping choose the correct sense-preserving lexical translation of ambiguous words. However, the same improvement in LTA score was also achieved with re-ranking using the Most Frequent Translation model that does not use the image. Human direct assessment preferred re-ranking with the Multimodal Word Sense Translation model and not the Most Frequent Translation model. In English-French, however, we get the opposite result where humans prefer re-ranking using the Most Frequent Translation model. In English-French, our strategy of re-ranking with Multimodal Word Sense Translation model does not impact METEOR and BLEU performances and helps improve LTA scores. This can be viewed as a positive outcome to some extent. In English-Czech, all re-ranking strategies decrease the performance of the Neural Machine Translation system drastically which can be attributed to the fact that our dataset of ambgiuous words and their lexical translations for English-Czech was noisy and hence the Sense Translation models were erroneous, thus leading to poor re-ranking. In the Multimodal and Multilingual Machine Translation task (Image + English + German + French $\longrightarrow$ Czech) our re-ranking approach was outperformed by the consensus based selection approach. In consensus based selection, a translation which appears in all three 10-best lists of English-Czech and German-Czech and French-Czech Neural Machine Translation systems is selected. In other words, this translation has been validated by all three systems.

Re-Ranking Strategy	METEOR ↑	BLEU ↑	TER ↓	LTA ↑	Human ↑
Image+English⟶German					
Baseline system* (no re-ranking)	47.4	27.6	55.2	45.3	67.4
Our NMT system (no re-ranking)	**50.7**	**30.9**	**52.4**	44.4	-
Re-ranked using:					
Most Frequent Translation	**50.7**	30.3	53.1	**48.3**	72.6
Text-only Word Sense Translation	**50.7**	30.5	53.0	48.0	-
Multimodal Word Sense Translation	**50.7**	30.4	52.9	**48.3**	**73.5**
Image+English⟶French					
Baseline system* (no re-ranking)	56.9	36.3	41.6	66.3	66.0
Our NMT system (no re-ranking)	**59.8**	**38.9**	**41.2**	67.9	-
Re-ranked using:					
Most Frequent Translation	59.7	38.8	41.6	67.6	**74.9**
Text-only Word Sense Translation	**59.8**	38.8	41.5	69.6	-
Multimodal Word Sense Translation	**59.8**	**38.9**	41.5	**69.9**	74.5
Image+English⟶Czech					
Baseline system* (no re-ranking)	27.7	26.5	54.4	62.1	57.8
Our NMT system (no re-ranking)	**29.4**	**29.0**	**51.1**	71.4	-
Re-ranked using:					
Most Frequent Translation	29.2	27.8	52.4	**73.6**	60.6
Text-only Word Sense Translation	29.1	28.3	51.7	72.1	-
Multimodal Word Sense Translation	29.1	28.2	51.7	71.4	**62.4**
Image+English+German+French⟶Czech					
Baseline system* (no re-ranking)	26.8	23.6	54.1	53.9	59.4
Re-ranking/selection using:					
Multimodal Word Sense Translation	27.5	24.5	52.5	**61.5**	**63.3**
Consensus based selection	**27.6**	24.7	**52.1**	**61.5**	-
Data Augmentation followed by Random Forest	27.1	24.1	54.6	51.9	-
Data Augmentation followed by RNN	27.5	**25.2**	53.9	51.9	61.8

Table 4.11: *Re-ranking n-best translations generated by a Neural Machine Translation system using Multimodal Word Sense Translation models. Systems evaluated on the Test 2018 dataset with METEOR, BLEU, TER, LTA metrics and Human Direct Assessment. * refers to system provided by Multimodal Machine Translation shared task organizers. Rest of the systems were developed by us. Highest scores are shown in* ***Bold Font****.*

In comparison, our re-ranking strategy using Multimodal Word Sense Translation model is only checking if ambiguous words in English are being translated correctly or not. It does not check for ambiguous words in German or French. Perhaps two more Multimodal Word Sense Translation models, one for German-Czech and one for French-Czech would have helped beat the consensus based selection strategy.

Finally, we wanted to understand why our approach of re-ranking n-best translations with Multimodal Word Sense Translation reduces the performance as measured using BLEU and TER despite increasing the performace as measure by LTA. On further inspection we found that the translations obtained using our strategy had other kind of errors like incorrect word order. These other errors go unnoticed by our strategy. Simply put, a translation of a source sentence has many qualities like fluency, reordering, grammar, lexical word choice, etc. Our strategy of re-ranking involves promoting a lower ranked translation based on just one quality (lexical word choice) while ignoring the other qualities. In the process, other qualities are compromised and with it the overall performance as measured by BLEU and TER drops. Despite the failure of our re-ranking strategy, we are still intrigued by the Oracle in Table 4.10. A re-ranking or a selection strategy capable of identifying the Oracle remains an interesting challenge which we would like to explore in future.

Chapter 5

Image Features and Word Embeddings for Multimodal Word Sense Translation

Image Features and Word Embeddings form the most basic units of 'tranfer learning' where knowledge acquired in one domain or task can be transferred and used in another domain or task. For Image Features, the standard practice is to extract these from a pre-trained Image Classification model trained on a massive corpus like the ImageNet which consists of more than 1.2 Million images (150 Gigabytes) in which objects have been annotated with 1000 object classes. For Word Embeddings, the standard practice is to extract these from a pre-trained Language Model trained on a giant corpus like the Google News dataset which consists of about 100 Billion words (2 Gigabytes). The knowledge acquired on these massive datasets about a word or an image is compressed down to vectors of a few hundred or thousand dimensions and then used in other domains and tasks as an input feature representing the word or the image. In this chapter, we explore the utility of these Image Features and Word Embeddings for our task of Multimodal Word Sense Translation.

5.1 Image Features

In the previous chapter, we learnt that by incorporating Image Features from the `pool5` layer of ResNet-50 Image Classification model into our model for Word Sense Translation improved the performance of our model in situations where the textual context was either compromised or only partially available or completely missing. Simply put, the Image Feature was compensating for the lack of textual context. Its contribution to our best performing Multimodal Word Sense Translation model, which was the Bidirectional Long Short-Term Memory network trained in the *ambiguous sentences* and *all words* data-setting, was inconclusive. Also, these Image Features in their raw form were not useful for Multimodal Word Sense Translation and had to undergo some transformation. For these reasons, we wondered whether these Image Features are really conducive for our task of Word Sense Translation? After taking closer look at the Image Classification task, we believe these Image Features may not be most conducive for our Task of Sense Translation for the following reasons:

- The Image Classification task is different from our intended task of Multimodal Word Sense Translation.
- The distribution of Images in ImageNet dataset is different from the distribution of Images in the 'Flickr30K' dataset from which we derived our dataset for Multimodal Word Sense Translation.
- The overlap of ambiguous words in our dataset and the 1000 object classes in ImageNet is poor. Only 12.41% of ambiguous words in English-French and 13.02% of ambiguous words in English-German are found in the object classes. Thus, the image features extracted may not be directly useful in disambiguating the remaining 87% of ambiguous words.
- For those ambiguous words which are found in the object classes (like the word 'hat'), the different sense-translations (i.e, the different kinds of hats which have different German words for them like 'hut', 'mutze', 'kappe', 'kopfbedeckung'. See Figure 3.6) are not found in the object classes. In other words, the Image Classification system is trained to identify all the different kinds of hats as one. It is not trained to classify between the different kinds of hats. So, features which are common in all the different kinds of hats are retained while the uncommon features which could help discern between the different kinds of hats may not be retained in the Image Features by the Image Classification model.

In an initial exploratory analysis of the Image Features, we extracted three sets of image pairs: (a) set A of image pairs where the two images in a pair have the same sense-translation label, (b) set B of image pairs where the two images in a pair correspond to the same ambiguous word but have different sense-translation labels, and, (c) set C of image pairs where the two images in a pair correspond to different ambiguous words. Note, we used ambiguous words and their sense-translation from our English-German dataset. Formally, let (x_1, y_1, v_1, t_1) and (x_2, y_2, v_2, t_2) be two samples from our dataset for Multimodal Word Sense Translation where x_* is an ambiguous word, y_* is its sense-translation label, v_* is the image and t_* is the textual context. Then, the set to which the image pair (v_1, v_2) belongs to is decided as follows:

$$\begin{aligned} &\text{if } y_1 = y_2, \text{ then } (v_1, v_2) \in A \\ &\text{if } y_1 \neq y_2 \text{ and } x_1 = x_2, \text{ then } (v_1, v_2) \in B \\ &\text{if } x_1 \neq x_2, \text{ then } (v_1, v_2) \in C \end{aligned} \tag{5.1}$$

Next, for every image pair (u, v) in these sets $(A, B$ and $C)$, we extracted their corresponding Image Features (u_f, v_f) from the `pool5` layer of a pre-trained ResNet-50 Image Classification model. Then, we computed the Cosine distance and the Euclidean distance (see Appendix A.13) between the Image Features in the pair denoted by $d_c(u_f, v_f)$ for Cosine distance and $d_e(u_f, v_f)$ for Euclidean distance. Finally, for each of the three sets, we computed the average of the Cosine distances and the average of the Euclidean distances.

For example, for set A we have the following:

$$\begin{aligned} \text{Average Cosine Distance of } A = A_c &= \frac{\sum_{(u,v)\in A} d_c(u_f, v_f)}{|A|} \\ \text{Average Euclidean Distance of } A = A_e &= \frac{\sum_{(u,v)\in A} d_e(u_f, v_f)}{|A|} \end{aligned} \tag{5.2}$$

Similarly, we computed B_c, B_e, C_c, and C_e. Next, for the Image Features to be conducive for our task of Multimodal Word Sense Translation, we expect the Image Features of images corresponding to the same sense-translation label to be closer to each other as compared to images with different sense-translation labels. In other words, we define Image Features to be conducive for our task if $A_c < B_c < C_c$ (and $A_e < B_e < C_e$) by a big enough margin. However, we found the following result in Table 5.1. Notice that $A_e > B_e$ and $A_c \approx B_c$ which violates our definition of Image Features being conducive for our task. This is in agreement to our previous observations that Image Features from ResNet-50 in their raw form are not useful for Multimodal Word Sense Translation and need to undergo some transformation to be useful. We propose another way to learn a transformation of the Image Features making them more conducive for our task using Siamese Network (Taigman et al. 2014) and Triplet Loss Function (Schroff et al. 2015) with the idea of bringing the Image Features corresponding to the same sense closer to each other and moving the Image Feutures corresponding to different senses further apart.

Set of image pairs	Cosine distance	Euclidean distance
Set A of image pairs having the same sense-translation label	0.32	20.6
Set B of image pairs of same ambiguous word but different sense	0.32	20.3
Set C of image pairs of different ambiguous words	0.38	25.8

Table 5.1: *Exploratory Analysis of Image Features from `pool5` layer of pre-trained ResNet-50 of the images in our dataset. We measured Cosine Distance and Euclidean Distance (see Appendix A.13) between Image Features of two images which either have the same sense-translation label (set A) or correspond to the same ambiguous word but with different sense-translation label (set B) or correspond to different ambiguous words (set C). We then took the average of the distances for each set.*

First, we extracted triplets of images (u, v, w) from our training set such that $(u, v) \in A$ corresponding to same sense-translation label and $(u, w) \in B$. We call (u, v) to be 'Same Sense Pair' and (u, w) to be 'Different Sense Pair'. In another version, we chose $(u, w) \in C$, but that didn't help us achieve much. For now, we will continue describing the first version.

Next, we built a simple fully connected Feed-Forward Neural Network like in Figure 4.1 but slightly different. The input layer of our network and the output layer of our network are both 2048-dimensional. We set the weight matrix W to be the Identity Matrix I with 1s along the diagonal and 0s everywhere else. The bias vector is all 0s. Thus, in this initial form, the Feed-Forward Neural Network will produce an output o for a given input feature

f as

$$o = W \cdot f + b = I \cdot f + b = f \tag{5.3}$$

In other words, this network does not do anything to the image feature fed into it in its current form.

Then, we pass the Image Features of u, v and w images obtained from the `pool5` layer of ResNet-50 (call it u_f, v_f and w_f respectively) into the Feed-Forward Neural Network to get three output vectors o_u, o_v and o_w respectively. Now, we define the Triplet Loss Function L as follows,

$$L(u, v, w) = \max(d(o_u, o_v) - d(o_u, o_w) + \alpha, 0) \tag{5.4}$$

where d is either the square of euclidean distance or the cosine distance and α is a hyperparameter called the 'margin'. Our cost function J is simply the mean of Triplet Loss Functions for all triplets in a batch as follows,

$$J_{\text{batch}} = \frac{\sum_{(u,v,w)\in\text{batch}} L(u, v, w)}{|\text{batch}|} \tag{5.5}$$

Finally, we backpropagate this cost/error function using garadient descent to learn new values of the parameters W and b. This learnt weights can be thought of as a transformation of the Image Features that makes them more conducive for our task of Multimodal Word Sense Translation. For hyperparameters, we had set the margin α to 0.1 if using Cosine distance or 2 if using square of Euclidean distance. The batch size was set to 32. We used ADAM optimizer with a learning rate of 0.0001 and trained our model for 2 epochs to avoid overfitting. Dropout was set to 0.3. Our model has about 4.2 Million parameters. Using our model described above, we transformed the ResNet-50 `pool5` Image Features of images in our dataset. We conducted the same Exploratory analysis of these transformed Image Features which we did on the original ResNet-50 `pool5` Image Features and got the following results in Table 5.2. We managed to satisfy our definition of 'Image Features conducive for our task of Multimodal Word Sense Translation'. However, when we used these transformed Image Features to initialise the Hidden State and the Cell State of the Bidirectional Long Short-Term Memory network model for our task of Multimodal Word Sense Translation, we did not

Set of image pairs	Cosine distance	Euclidean distance
Transformed Image Features where loss function used Cosine distance		
Set A of image pairs having the same sense-translation label	0.29	19.3
Set B of image pairs of same ambiguous word but different sense	0.33	20.1
Set C of image pairs of different ambiguous words	0.38	26.4
Transformed Image Features where loss function used squared Euclidean distance		
Set A of image pairs having the same sense-translation label	0.32	19.7
Set B of image pairs of same ambiguous word but different sense	0.33	22.1
Set C of image pairs of different ambiguous words	0.38	25.9

Table 5.2: *Exploratory Analysis of the transformed Image Features. The original Image Features from `pool5` layer of pre-trained ResNet-50 of the images in our dataset were transformed by our Feed-Forward Neural Network trained using Triplet loss in Equation 5.4.*

see any improvement over using the original Image Features from `pool5` layer of ResNet-50 model (see Table 5.3). We speculate, this is because the network was already transforming the original image features internally and it does not need any additional transformation of the Image Feature externally.

Model	DS 1	DS 2	DS 3	DS 4
Most Frequent Translation	65.3	65.3	65.3	65.3
BLSTM (text-only)	67.6	68.2	69.0	69.6
BLSTM + original Image Features	68.4 (+0.8)	68.6 (+0.4)	68.7 (-0.3)	69.8 (+0.2)
BLSTM + transformed^C Image Features	68.2 (+0.6)	67.6 (-0.6)	68.4 (-0.6)	69.2 (-0.4)
BLSTM + transformed^E Image Features	67.4 (-0.2)	68.2 (0.0)	69.9 (+0.9)	69.1 (-0.5)
BLSTM + original objects vector	67.8 (+0.2)	69.0 (+0.8)	69.9 (+0.9)	69.8 (+0.2)
BLSTM + detected objects vector	68.3 (+0.7)	68.6 (+0.4)	69.0 (0.0)	69.1 (-0.5)
BLSTM + prepend original objects	70.1 (+2.5)	70.4 (+2.2)	70.9 (+1.9)	71.0 (+1.4)
BLSTM + prepend detected objects	65.7 (+1.1)	69.5 (+1.3)	69.8 (+0.8)	69.7 (+0.1)

Table 5.3: *Image Features for Multimodal Word Sense Translation. We have a Bidirectional Long Short-Term Memory (BLSTM) network and we initialize its Hidden State and Cell State with different Image Features. These include (a) the original Image Features which are obtained from `pool5` layer of pre-trained ResNet-50, (b) transformed^C Image Features where original Image Features are transformed using triplet loss with Cosine distance, (c) transformed^E Image Features where original Image Features are transformed using Triplet loss with squared Euclidean disatance, (d) original object vectors which is a vector of objects in the image, (e) detected object vector which is a vector of objects detected by the Object Detection model, (f) prepend * objects which refers to pre-pending object classes to the textual context. DS stands for Data-setting. DS 1 is 'all sentence + ambiguous words' data-setting. DS 2 is 'ambiguous sentence + ambiguous words' data-setting. DS 3 is 'all sentence + all words' data-setting. DS 4 is 'ambiguous sentence + all words' data-setting.*

Next, we thought of going to a higher level of abstraction (see Figure 2.6) by using explicit objects in the image. The ambiguous words in our dataset are content words, usually nouns, which we assume correspond better to the objects or object classes in the image. So we propose using objects in the image as a Representation of the Image (Image Features). We extracted objects from the image in two ways: (a) Human annotations and (b) Object detection model.

1. Human annotations: The images in our dataset (except Test 2017 MSCOCO) are originally from Flickr30K which are annotated with object classes as described in Plummer et al. (2015). Since these object classes are annotated by humans, we also call it 'original objects' in the image.

2. Object detection model: we used the pre-trained 'bottom-up-attention' Object Detection model in Anderson et al. (2018). We extracted 10 objects from each image using this model. Since these objects are detected by an Object Detection model and not annotated by humans, we call these 'detected objects'.

Next, to represent the 'original objects' and 'detected objects' in the image, we tried two kinds of representations:

1. Bag-of-Objects: Inspired by Wang et al. (2018), we created bag-of-object vectors for every image in our dataset. Formally, if $O = \{o_1, o_2, ..., o_{|O|}\}$ are the object classes then given an image I, its Bag-of-Objects representation is the vector $(v_1, v_2, ..., v_{|O|})$, such that
$$v_i = \begin{cases} 1, & \text{if } o_i \in I \\ 0, & \text{if } o_i \notin I) \end{cases} \tag{5.6}$$

2. Objects as words: Motivated by the success of prepending language class label as words to the source sentence in Google's Multilingual Neural Machine Translation system (Johnson et al. 2017), we simply consider the Object labels as words in the English vocabulary and then prepend it to the textual context as depicted in Figure 5.1.

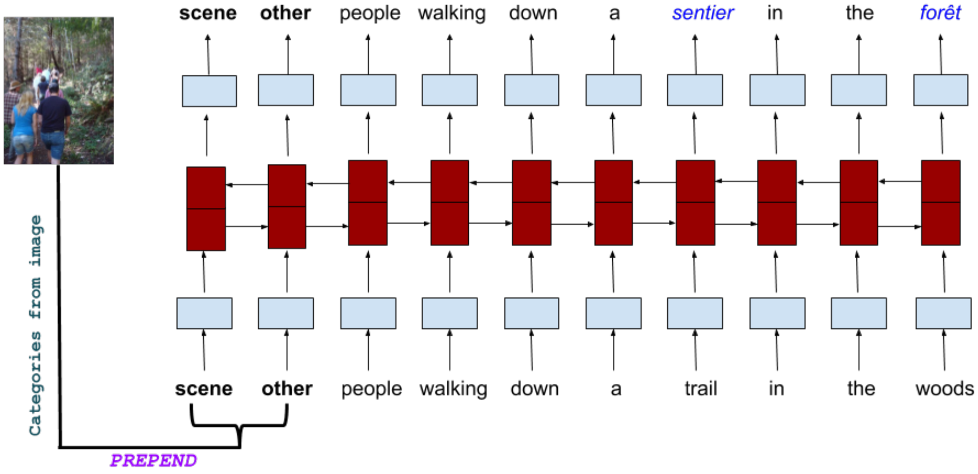


Figure 5.1: *Objects or Object categories in the image are prepended to the textual context. A Bidirectional Long Short-Term Memory Network then reads the objects or object categoires as words in the English vocabulary. We have shown for the all words configuration in this figure. For all words configuration, please refer Figure 4.2.*

The results are presented in Table 5.3. We observe that our Multimodal Word Sense Translation model clearly benefits from the objects or object categories found in the image. More specifically, the object categories annotated by humans in Plummer et al. (2015) seem to be more useful for our task. Also, the approach of using objects as words in the English vocabulary followed by prepending these to the textual context seems to play a significant role in improving the performance of our model as compared to creating a vector representation of objects and then initializing the Hidden State and Cell State of the Long Short-Term Memory units. We believe this is because by treating object classes as words in English Vocabulary, we are embedding the object from the visual space directly into the textual semantic space. In previous approaches, Image features and Word Embeddings were in different spaces (Visual space was different from textual semantic space) and the model had to work towards effectively combining the two. In recent times, this approach is being adopted in other works like in the state-of-the-art OSCAR model (Li et al. 2020).

5.2 Word Embeddings

Use of pre-trained word embeddings from a Language Model in a downstream task is a common practice in Natural Langugae Processing. We would like to know the utility of such pre-trained word embeddings in our task of Multimodal Word Sense Translation. We experiment with four popularly used word embeddings, (1) Word2Vec (Mikolov, Sutskever, Chen, Corrado & Dean 2013), (2) GloVe (Pennington et al. 2014), (3) ELMo (Peters et al. 2018), and (4) BERT (Devlin et al. 2018). In addition to these, since our dataset is multimodal, we also consider Multimodal Embeddings obtained from a BERT-like model VL-BERT (Su et al. 2020).

- **Word2Vec (Mikolov, Sutskever, Chen, Corrado & Dean 2013)** is a Feed-Forward Neural Network Language Model which comes in two flavours - (1) Continuous Bag-of-Words (CBOW) model and (2) Continuous Skip-gram model - which have been depicted in the Figure 5.2 below. In our work, we used the Skip-gram variant of Word2Vec because it is known to perform better at capturing semantic information. The Word2Vec embeddings have been trained on a corpus of over 6 Billion words and we would like to see if the semantic information about words gained from this massive training corpus improves our Word Sense Translation models or not.

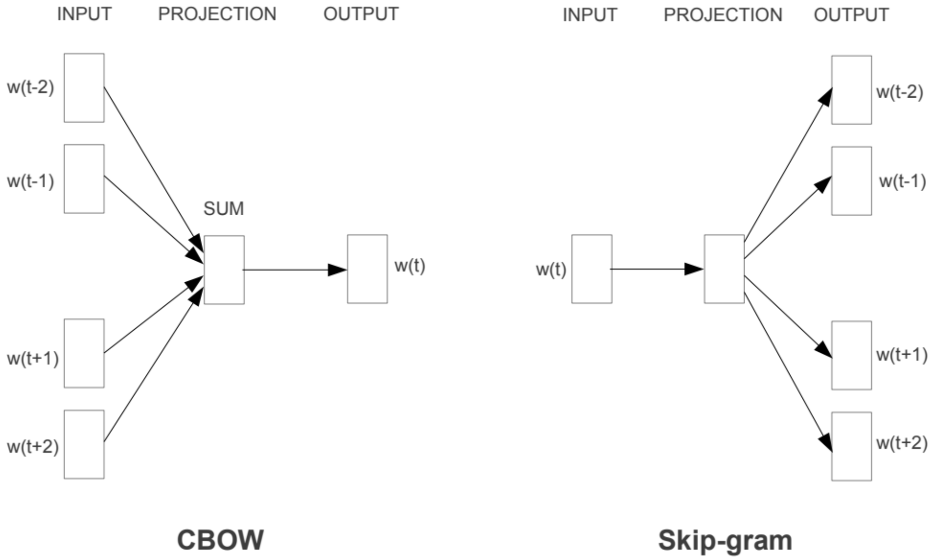


Figure 5.2: *Two variants of Word2Vec - Continuous Bag-of-Words (CBOW) and Skip-gram. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.*

- **GloVe (Pennington et al. 2014)** embeddings utilise word co-occurence matrix built from a corpus of 6 Billion words (same corpus used in Word2Vec). The GloVe model is trained on the non-zero entries of this global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. Populating this matrix requires a single pass through the entire corpus to collect the statistics. The model is essentially a log-bilinear model with a weighted least-squares objective. The main intuition underlying the model is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. It is regarded as one of the best word embeddings from a statistical (non-neural network) language models.

- **ELMo (Peters et al. 2018)** is a contextualised word embedding obtained from a deep Bidirectional Long Short-Term memory network (similar to our BLSTM model architecture). This model has two layers of Bidirectional LSTM units and it is trained on the One Billion Word Benchmark corpus (Chelba et al. 2013). It's first layer representation is taken as a word embedding. Unlike Word2Vec and GloVe, this is a contextualised word embeddings, as it generates a different embeddings of the same word in different contexts, and hence it is capable of capturing the different senses of an ambiguous word from the context. Also, since the model architecture is identical to our BLSTM model, we believe it is a good starting point to experiment with contextualised embeddings.

- **BERT (Devlin et al. 2018)** is another contextualised word embedding which utilizes the Transformer architecture as against the LSTM architecture used in ELMo. The transformer architecture is a novel neural network architecture which has many benefits over the conventional sequential models (LSTM, RNN, GRU etc) which include, but are not limited to, the more effective modeling of long term dependencies among tokens in a temporal sequence, and the more efficient training of the model in general by eliminating the sequential dependency on previous tokens. It is basically an encoder-decoder architecture model which uses attention mechanisms to forward a more complete picture of the whole sequence to the decoder at once rather than sequentially in an LSTM. Central to the transformer model is the notion of self-attention

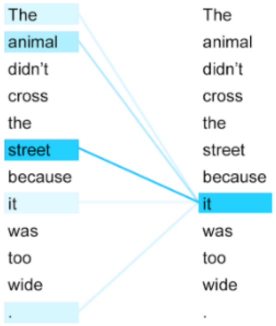


Figure 5.3: *Self-attention example. Meaning of a word depends on every word in the context to different extents which is reflected in the attention weights. The meaning of the word 'it' is more dependent on 'street' as compared to other words.*

depicted in Figure 5.3. The self-attention mechanism assumes that the meaning of each word depends on every word in the context including itself. The only question is to what extent? This is captured in the attention weights which signify some words in the context are more important than others. For example, the word 'it' in the figure is more dependent on the word 'street' so its corresponding attention weight will be more. This is ideal for capturing the sense of an ambiguous word where the sense may depend on words which could be anywhere in the context. The self-attention mechanism can be extended to every word with multiple heads and then further to many layers eventually forming a complex model architecture depicted in Figure 5.4. For a more technical understanding of Transformer architecture, please refer to Vaswani et al. (2017) and Devlin et al. (2018).

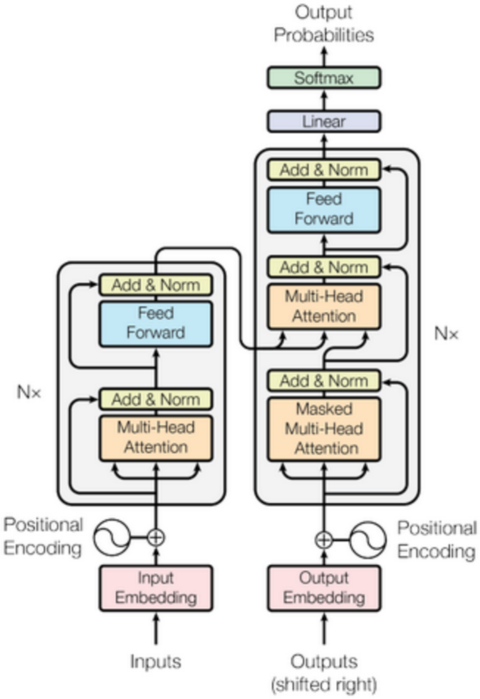


Figure 5.4: *The Transformer architecture used to get the BERT embeddings.*

- **VL-BERT (Su et al. 2020)** is one of the many multimodal extensions of BERT which has gained popularity in recent times. The VL-BERT architecture is depicted in the Figure 5.5. The backbone of VL-BERT is of (multi-modal) Transformer attention module taking both visual and linguistic embedded features as input. In it, each element is either of a word from the input sentence, or a region-of-interest (RoI) from the input image, together with certain special elements to disambiguate different input formats. Each element can adaptively aggregate information from all the other elements according to the compatibility defined on their contents, positions, categories, and etc. The content features of an RoI are domain specific (Fast R-CNN (Girshick et al. 2014) features). By stacking multiple layers of multi-modal Transformer attention modules, the derived representation is of rich capability in aggregating and align-

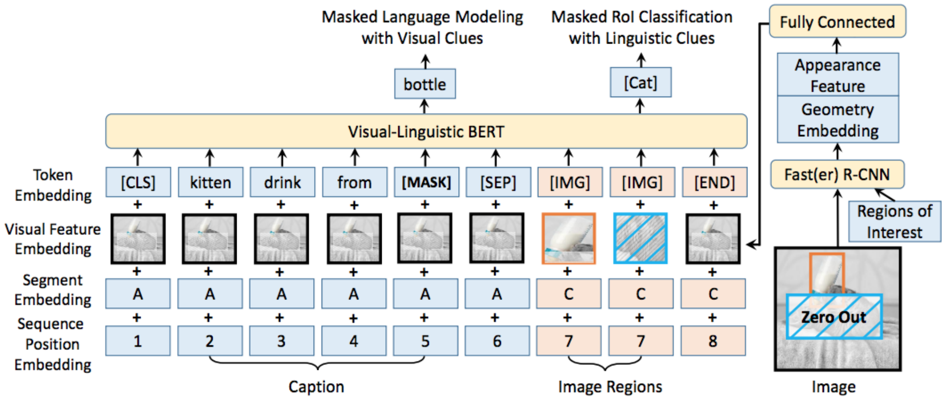


Figure 5.5: *The Transformer architecture of Visual-Liguistic BERT.*

ing visual-linguistic clues. Also, task-specific branches can be added above for specific visual-linguistic tasks. VL-BERT is trained on large-scale visual-linguistic corpus of Conceptual Captions Dataset (Sharma et al. 2018). We want to check if pre-trained Multimodal extensions of BERT can enhance the performance of our downstream task of Multimodal Word Sense Translation.

We used the above word embeddings and multimodal embeddings in the Embedding layer of our Multimodal Word Sense Translation model (see Figure 4.5). The results are presented in Table 5.4.

Model	DS 1	DS 2	DS 3	DS 4
Most Frequent Translation	65.3	65.3	65.3	65.3
BLSTM	67.6	68.2	69.0	69.6
BLSTM + Word2Vec	66.7 (-0.9)	68.8 (+0.6)	68.9 (-0.1)	69.9 (+0.3)
BLSTM + GloVe	67.9 (+0.3)	68.4 (+0.2)	68.9 (-0.1)	69.6 (0.0)
BLSTM + ELMo	68.6 (+1.0)	69.3 (+1.1)	69.8 (+0.9)	69.5 (-0.1)
BLSTM + BERT	69.2 (+1.6)	69.2 (+1.0)	69.5 (+0.5)	70.1 (+0.5)
BLSTM + VL-BERT	68.4 (+0.8)	69.0 (+0.8)	69.5 (+0.5)	69.8 (+0.2)
BLSTM + prepend original objects	70.1 (+2.5)	70.4 (+2.2)	70.9 (+1.9)	71.0 (+1.4)
BLSTM + prepend original objects + BERT	69.4 (+1.8)	70.5 (+2.3)	70.1 (+1.1)	70.9 (+1.3)

Table 5.4: *Word Embeddings for Multimodal Word Sense Translation. We have a Bidirectional Long Short-Term Memory network (see Figure 4.5) and we initialize the Embeddings layer with Word Embeddings like Word2Vec, GloVe, ELMo, BERT and VL-BERT. We also provide the best performing model which is BLSTM + prepend original objects from Table 5.3. We used BERT on top of this and found it does not improve the performance any further. DS stands for Data-setting. DS 1 is 'all sentence + ambiguous words' data-setting. DS 2 is 'ambiguous sentence + ambiguous words' data-setting. DS 3 is 'all sentence + all words' data-setting. DS 4 is 'ambiguous sentence + all words' data-setting.*

We observe that Word Embeddings, in general, do seem to benefit Word Sense Translation. The improvements from Word2Vec and GloVe are marginal and hence their impact seems inconclusive. We believe this is because these are non-contextual embeddings, i.e a word gets the same embedding irrespective of the context, and hence unable to capture 'sense' information of a word. ELMo and BERT on the other hand are contextualized Embeddings and these are clearly useful for our task. VL-BERT is a contextualized multimodal embedding and it also improves upon the baseline model, however the performance gains do not surpass the text-only BLSTM with Word Embeddings from BERT. Despite the gains from contextualized word embeddings, our best performing model continues to be the one where object categories in the image are prepended as words to the text textual context. We explored the option of using BERT embeddings on top of this best performing model, denoted as 'BLSTM + prepend original objects + BERT', but this does not improve the performance of the best performing model any further. Finally, we would like to highlight the importance of Data-setting. It seems, we get more gains from the Word Embeddings in the 'ambiguous words' data-settings (DS 1 and DS 2) where unambiguous words are tagged to a common label underscore '_'. We had previously discussed that in such a data-setting, the Long Short-Term Memory units are, perhaps, not capturing the textual context better. Word embeddings seem to compensate for this compromise in the representation of the textual context.

Chapter 6

Conclusions

In this thesis, we introduced the task of Multimodal Word Sense Translation where the objective is to translate an ambiguous word given a textual context and a visual context. Our task was inspired by Multimodal Machine Translation and Visual Sense Disambiguation.

We carried out a thorough and detailed review of literature relevant to our task beginning with Machine Translation. We reviewed the two prominent paradigms of Machine Translation which are Statistical Machine Translation and Neural Machine Translation. We looked at the problems in Machine Translation and identified 'Lexical Choice Errors', like translation of Homographs and Polysemes and transfer ambiguities, to be a major challenge in improving translation quality. We then reviewed the task of Word Sense Disambiguation and more specifically the cross-lingual variant of Word Sense Disambiguation. Next, we reviewed the previous attempts of utilizing Word Sense Disambiguation models in Machine Translation and found that these have not been easy to integrate. We realised, both Machine Translation and Word Sense Disambiguation would benefit from additional contextual information. One way to add additional contextual information is to look for it in other modalities like vision. We looked at the recent success of Deep Convolutional Neural Networks at Image Classification and reviewed its ability to capture visual context in different 'levels or spectrum of abstraction' (see Figure 2.6). We noticed the overlap between Computer Vision and Natural Language Processing in terms of objects (nouns) in the level of abstraction. Next, we reviewed a few Language and Vision tasks which are relevant to our task like Image Captioning, Multimodal Machine Translation and Word Sense Disambiguation. We also explored the different approaches and strategies adopted to solve these problems and were inspired to develop our models based on these ideas.

We then formulated our dataset as a 4-tuple of an ambiguous word, its sense-preserving lexical translation, its textual context which is a sentence and its visual context which is an image. We created our dataset from Multi30K, an existing dataset for Multimodal Machine Transltion, using word alignment followed by human filtering. We analysed our dataset and the human filtering process in extensive detail and noticed that the ambiguity of a word in terms of its lexical translations is not easy to define or quantify. Based on the statistics of our dataset, we quantified the extent of ambiguity in terms of the skewness of its sense-translation labels. We explored the human annotation process and found that humans prefer to look at images for translation when they encounter a polyseme or a transfer ambiguity (and not for category ambiguities or homographs). Next, we used our dataset to measure the Multimodal Word Sense Translation capabilities of the systems submitted to the Multimodal Machine Translation shared task. Our metric of sense-preserving Lexical Translation Accuracy was useful in analysing the submitted systems and hence, it was widely appreciated by the Multimodal Machine Translation community.

Next, we developed models for Multimodal Word Sense Translation. We noticed that a very basic Most Frequent Translation of an ambiguous word could be very strong baseline model for our task due to the skewed dataset. Other simple baseline models included k-Nearest Neighbours and a single layered Feed-Forward Neural Network. In experiments with these basic models, we realised that Image Features obtained from `pool5` layer of ResNet-50 in its raw form is not useful for Sense Translation and it needs to undergo some transformation to be useful. We also realised that, because a sentence can have multiple ambiguous words, we need to transform the Sense translation problem from classification (like in Word Sense Disambiguation) to sequence tagging. We developed sequence tagging models for our task using Long Short-Term Memory networks. Our best performing sequence tagging model is a Bidirectional Long-Short Term Memory network. We experimented with incorporating Image Features from ResNet-50 into our models by initialising the Hidden State and the Cell State of the Long Short-Term Memory units with the Image Features. We found Image Features to be useful for Unidirectional Long Short-Term Memory networks. In general, we noticed Image Features are beneficial for our task whenever textual context is either compromised or partially available or missing.

Next, we wanted to use our Multimodal Word Sense Translation models to improve the translation quality of a Machine Translation system. We noticed that the n-best translation outputs of a Machine Translation system often has a translation (we call it the Oracle) which is better than the 1-best output. In fact, the Oracle of 20-best translation output of a Neural Machine Translation system can outperform the 1-best translation output by upto 13.5 METEOR points. So, we tried to re-rank the 10-best translation outputs using our Sense Translation models in an attept to spot the Oracle translation. We found that our re-ranking strategy improved the lexical translation of ambiguous words, but it came at the cost of other translation errors like re-ordering.

Next, we defined what it means for Image Features to be conducive for our task in terms of distance between Image Features corresponding to the same 'sense' versus the distance between Image Features corresponding to different 'senses'. We found that the popularly used ResNet-50 Pooled features are not conducive for our task of Multimodal Word Sense Translation based on this definition. So we proposed a way of transforming the Image Features to make these conducive using Siamese Network and Triplet Loss function. However, despite the transformation achieving the criteria of being conducive, the new Image Features we obtained did not improve the performance of our models as compared to the original untransformed Image Features. This shows that the Bidirectional Long Short-Term Memory network is quite capable of transforming Image Features internally for its use and does not need an external transformation the way we did. Next, we experimented with representing an image by the objects found in it. We found that using objects in an image as words in English vocabulary is the most beneficial representation of the image for our task. This is because we are embedding the object (or object categories) in the image directly into the textual semantic space so the model, in essence, does not have to deal with information in two different modalities. We also explored various popular Word Embeddings for our task and found that contextualised word embeddings like ELMo and BERT improve the performance of Word Sense Translation model. However Multimodal Contextualised embeddings like VL-BERT do not improve the performance of our model any further.

6.1 Future Work

In this PhD, we have identified several avenues and problems for further research. These include the following,

- **Dataset for Multimodal Word Sense Translation:** The current dataset has several problems like Skewed distribution of sense translations, lack of 'visually ambiguous' words, etc. We would like to address these problems in future and develop a more challenging dataset with more polysemes and transfer ambiguities, and, a uniform distribution of sense-translations so that our dataset is less biased. One proposed strategy is to first identify a list of visually ambiguous words which we expect will benefit from an accompanying image and then extract an image and a description for the visually ambiguous words via crowdsourcing or by some automated means. This is opposite to the current approaches of data collection where we first extract images and then create its description and translations. As a result, we end up with very few visually ambiguous words in our dataset.

- **Models for Multimodal Word Sense Translation:** We have struggled to keep pace with the fast moving field of Machine Learning and Deep Learning, hence, we have not been able to develop Transformer-based models for our task. As discussed in section 5.2 when we covered BERT and VL-BERT, the Transformer architecture has a number of advantages over the sequential architectures, the most prominent of which is effective modelling of long term dependency. In a multimodal setting, we would like to take advantage of effective modelling of long-term dependency of a visually ambiguous word over the Image features. The main strength of the Transformer lies in layers of 'self-attention' that enable the model to gather a higher levels of abstraction. For a visually ambiguous word, attention over image features followed by multiple layers of self-attention can prove to be useful in identifying the sense of the ambiguous word in the given context. Thus multimodal transformer architectures like the one used in VL-BERT (see Figure 5.5) may be relevant for our task of Multimodal Word Sense Translation where instead of Masked Language modeling task, we perform Word Sense Translation of a masked ambiguous word. Many experiments can be conducted with respect to the kind of image features to use. Thus, in future, we would like to explore more model architectures like multimodal transformer architecture used in VL-BERT suitable for our task.

- **The Oracle in n-best translations:** We are intrigued by the Oracle in n-best translations generated by a Machine Translation system. We are surprised why the gap between the Oracle and the 1-best translation is so big and why no one has managed to explain this variance. We would like to pursue this topic in the future because it can potentially improve the performance of Machine Translation systems by a huge margin.

- **Image Features for Multimodal Word Sense Translation:** We believe, for a more useful merger between Computer Vision and Natural Language Processing, the Image Features need to capture higher levels of abstraction beyond objects (nouns) in the image. We could potentially use the Multimodal Word Sense Translation task to improve Computer Vision at identifying higher levels of abstraction from the images.

- **Multimodal Embeddings:** Use of transformer architecture to process information in multiple modalities and then use the hidden states of the trained transformer for downstream tasks has been a recent trend. Making it more effective by having Multimodal Word Sense Translation as one of its tasks would be potential research avenue. In section 5.2, we used a pre-trained multimodal embedding for Multimodal Word Sense Translation. In future we would like to see if we can reverse the process and use the task of Multimodal Word Sense Translation to improve the quality of the pre-trained multimodal embeddings.

Bibliography

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M. et al. (2016), Tensorflow: A system for large-scale machine learning, *in* '12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)'.

Agirre, E. & Edmonds, P. (2007), *Word Sense Disambiguation: Algorithms and Applications*, Springer Science & Business Media.

Agirre, E. & Soroa, A. (2007), Semeval-2007 task 02: Evaluating word sense induction and discrimination systems, *in* 'Proceedings of the 4th International Workshop on Semantic Evaluations', Association for Computational Linguistics, pp. 7–12.

Alm, C. O., Loeff, N. & Forsyth, D. (2006), Challenges for annotating images for sense disambiguation, *in* 'Proceedings of the Workshop on Frontiers in Linguistically Annotated Corpora 2006', pp. 1–4.

ALPAC (1966), *Language and Machines: Computers in Translation and Linguistics; a Report*, National Academy of Sciences, National Research Council.

Anderson, P., Fernando, B., Johnson, M. & Gould, S. (2016), Spice: Semantic propositional image caption evaluation, *in* 'European Conference on Computer Vision', Springer, pp. 382–398.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S. & Zhang, L. (2018), Bottom-up and top-down attention for image captioning and visual question answering, *in* 'CVPR'.

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C. & Parikh, D. (2015), Vqa: Visual question answering, *in* 'Proceedings of the IEEE international conference on computer vision'.

Bahdanau, D., Cho, K. & Bengio, Y. (2015), Neural machine translation by jointly learning to align and translate, *in* 'International Conference on Learning Representations'.

Banerjee, S. & Pedersen, T. (2002), An adapted lesk algorithm for word sense disambiguation using wordnet, *in* 'International Conference on Intelligent Text Processing and Computational Linguistics', Springer.

Bannard, C. & Callison-Burch, C. (2005), Paraphrasing with bilingual parallel corpora, *in* 'Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics', Association for Computational Linguistics, pp. 597–604.

Bar-Hillel, Y. (1964), *Language and Information*, Addison-Wesley.

Barnard, K. & Johnson, M. (2005), 'Word sense disambiguation with pictures', *Artificial Intelligence* .

Barrault, L., Bougares, F., Specia, L., Lala, C., Elliott, D. & Frank, S. (2018), Findings of the third shared task on multimodal machine translation, *in* 'Proceedings of the Third Conference on Machine Translation: Shared Task Papers', Association for Computational Linguistics.

Basile, P., Caputo, A. & Semeraro, G. (2014), An enhanced lesk word sense disambiguation algorithm through a distributional semantic model, *in* 'Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers'.

Benesty, J., Chen, J., Huang, Y. & Cohen, I. (2009), Pearson correlation coefficient, *in* 'Noise reduction in speech processing', Springer, pp. 37–40.

Bentivogli, L., Bisazza, A., Cettolo, M. & Federico, M. (2016), Neural versus phrase-based machine translation quality: a case study, *in* 'Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics.

Bernardi, R., Cakici, R., Elliott, D., Erdem, A., Erdem, E., Ikizler-Cinbis, N., Keller, F., Muscat, A. & Plank, B. (2016), 'Automatic description generation from images: A survey of models, datasets, and evaluation measures', *Journal of Artificial Intelligence Research* .

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L. & Turchi, M. (2015), Findings of the 2015 workshop on statistical machine translation, *in* 'Proceedings of the Tenth Workshop on Statistical Machine Translation', Association for Computational Linguistics.

Bojar, O., Federmann, C., Fishel, M., Graham, Y., Haddow, B., Huck, M., Koehn, P. & Monz, C. (2018), Findings of the 2018 conference on machine translation (wmt18), *in* 'Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers', Association for Computational Linguistics, Belgium, Brussels.

Brown, P. F., Cocke, J., Della Pietra, S. A., Della Pietra, V. J., Jelinek, F., Mercer, R. L. & Roossin, P. (1988), A statistical approach to language translation, *in* 'CoLing Budapest 1988 Volume 1: International Conference on Computational Linguistics'.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J. & Mercer, R. L. (1991), Word-sense disambiguation using statistical methods, *in* '29th Annual meeting of the Association for Computational Linguistics'.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D. & Mercer, R. L. (1993), 'The mathematics of statistical machine translation: Parameter estimation', *Comput. Linguist.* .

Cabezas, C. & Resnik, P. (2005), Using wsd techniques for lexical selection in statistical machine translation, Technical report, Maryland University College Park Institute for Advanced Compuer Studies.

Caglayan, O., Aransa, W., Bardet, A., Garcia-Martinez, M., Bougares, F., Barrault, L., Masana, M., Herranz, L. & van de Weijer, J. (2017), Lium-cvc submissions for wmt17 multimodal translation task, *in* 'Second Conference on Machine Translation'.

Caglayan, O., Aransa, W., Wang, Y., Masana, M., García-Martínez, M., Bougares, F., Barrault, L. & van de Weijer, J. (2016), 'Does multimodality help human and machine for translation and image captioning?', *In proceedings of the First Conference on Machine Translation* .

Caglayan, O., García-Martínez, M., Bardet, A., Aransa, W., Bougares, F. & Barrault, L. (2017), 'Nmtpy: A flexible toolkit for advanced neural machine translation systems', *In proceedings of The Prague Bulletin of Mathematical Linguistics* .

Caglayan, O., Madhyastha, P., Specia, L. & Barrault, L. (2019), Probing the need for visual context in multimodal machine translation, *in* 'Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)', Association for Computational Linguistics.

Calixto, I., de Campos, T. & Specia, L. (2012), 'Images as context in statistical machine translation', *The 2nd Annual Meeting of the EPSRC Network on Vision Language (VL12), Sheffield, UK. EPSRC Vision and Language Network.* .

Calixto, I., Dutta Chowdhury, K. & Liu, Q. (2017), Dcu system report on the wmt 2017 multi-modal machine translation task, *in* 'Second Conference on Machine Translation'.

Calixto, I., Liu, Q. & Campbell, N. (2017), 'Doubly-attentive decoder for multi-modal neural machine translation', *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* .

Callison-Burch, C., Osborne, M. & Koehn, P. (2006), Re-evaluating the role of bleu in machine translation research, *in* 'In EACL', pp. 249–256.

Carpuat, M. (2013), Nrc: A machine translation approach to cross-lingual word sense disambiguation (semeval-2013 task 10), *in* 'Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)'.

Carpuat, M. & Wu, D. (2005), Word sense disambiguation vs. statistical machine translation, *in* 'Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)'.

Carpuat, M. & Wu, D. (2007), Improving statistical machine translation using word sense disambiguation, *in* 'Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)'.

Chan, Y. S., Ng, H. T. & Chiang, D. (2007), Word sense disambiguation improves statistical machine translation, *in* 'Proceedings of the 45th annual meeting of the association of computational linguistics'.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P. & Robinson, T. (2013), 'One billion word benchmark for measuring progress in statistical language modeling', *INTERSPEECH* .

Chen, X., Ritter, A., Gupta, A. & Mitchell, T. (2015), Sense discovery via co-clustering on images and text, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition'.

Cho, K., van Merriënboer, B., Bahdanau, D. & Bengio, Y. (2014), On the properties of neural machine translation: Encoder–decoder approaches, *in* 'Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation', Association for Computational Linguistics.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014), Empirical evaluation of gated recurrent neural networks on sequence modeling, *in* 'NIPS 2014 Workshop on Deep Learning, December 2014'.

Clough, P., Grubinger, M., Deselaers, T., Hanbury, A. & Müller, H. (2006), Overview of the imageclef 2006 photographic retrieval and object annotation tasks, *in* 'Workshop of the Cross-Language Evaluation Forum for European Languages', Springer.

Dale, R., Moisl, H. & Somers, H. (2000), *Handbook of Natural Language Processing*, CRC Press.

Davies, D. L. & Bouldin, D. W. (1979), 'A cluster separation measure', *IEEE transactions on pattern analysis and machine intelligence* .

Delbrouck, J. & Dupont, S. (2017), 'An empirical study on the effectiveness of images in multimodal neural machine translation', *Computing Research Repository* **abs/1707.00995**.

Dempster, A. P., Laird, N. M. & Rubin, D. B. (1977), 'Maximum likelihood from incomplete data via the em algorithm', *Journal of the Royal Statistical Society: Series B (Methodological)* **39**(1), 1–22.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009), ImageNet: A Large-Scale Hierarchical Image Database, *in* 'Conference on Computer Vision and Pattern Recognition'.

Deng, J., Russakovsky, O., Krause, J., Bernstein, M., Berg, A. C. & Fei-Fei, L. (2014), Scalable multi-label annotation, *in* 'ACM Conference on Human Factors in Computing Systems (CHI)'.

Denkowski, M. & Lavie, A. (2014), Meteor universal: Language specific translation evaluation for any target language, *in* 'Proceedings of the Ninth Workshop on Statistical Machine Translation'.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018), 'Bert: Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv:1810.04805* .

Djiako, G. A. (2019), *Lexical Ambiguity in Machine Translation and its Impact on the Evaluation of Output by Users*, Saarland University Press.

Dunn, J. C. (1973), 'A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters', *Cybernetics* .

Dunn, J. C. (1974), 'Well-separated clusters and optimal fuzzy partitions', *Journal of cybernetics* .

Dyer, C., Chahuneau, V. & Smith, N. A. (2013), A simple, fast, and effective reparameterization of IBM model 2, *in* 'Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2013)'.

Edmonds, P. (2002), 'Senseval: The evaluation of word sense disambiguation systems', *European Language Resources Association (ELRA) newsletter* .

Elliott, D., Frank, S., Barrault, L., Bougares, F. & Specia, L. (2017), Findings of the Second Shared Task on Multimodal Machine Translation and Multilingual Image Description, *in* 'Proceedings of the Second Conference on Machine Translation'.

Elliott, D., Frank, S. & Hasler, E. (2015), 'Multilingual image description with neural sequence models', *Computing Research Repository arXiv:1510.04709* .

Elliott, D., Frank, S., Sima'an, K. & Specia, L. (2016), Multi30k: Multilingual english-german image descriptions, *in* 'Proceeding of the 5th Workshop on Vision and Language'.

Elliott, D. & Kádár, Á. (2017), Imagination improves multimodal translation, *in* 'Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)', pp. 130–141.

Elman, J. L. (1990), 'Finding structure in time', *Cognitive Science* .

Escudero, G., Màrquez, L. & Rigau, G. (2000), Naive bayes and exemplar-based approaches to word sense disambiguation revisited, *in* 'Proceedings of the 14th European Conference on Artificial Intelligence, ECAI', Citeseer.

Farhadi, A., Hejrati, M., Sadeghi, M. A., Young, P., Rashtchian, C., Hockenmaier, J. & Forsyth, D. (2010), Every picture tells a story: Generating sentences from images, *in* 'European Conference on Computer Vision', Springer.

Federmann, C. (2012), 'Appraise: An open-source toolkit for manual evaluation of machine translation output', *The Prague Bulletin of Mathematical Linguistics* .

Fellbaum, C. (1998), *WordNet*, Wiley Online Library.

Fellbaum, C. (2012), 'Wordnet', *The Encyclopedia of Applied Linguistics* .

Firth, J. R. (1957), 'Applications of general linguistics', *Transactions of the Philological Society* .

Fomicheva, M. & Specia, L. (2016), Reference bias in monolingual machine translation evaluation, *in* '54th Annual Meeting of the Association for Computational Linguistics', ACL.

Frege, G. (1892), 'On sense and reference [german: Über sinn und bedeutung]', *Zeitschrift für Philosophie und Philosophische Kritik* .

Freitag, M. & Al-Onaizan, Y. (2017), Beam search strategies for neural machine translation, *in* 'Proceedings of the First Workshop on Neural Machine Translation', Association for Computational Linguistics.

Gella, S., Lapata, M. & Keller, F. (2016), Unsupervised visual sense disambiguation for verbs using multimodal embeddings, *in* 'Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)'.

Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014), Rich feature hierarchies for accurate object detection and semantic segmentation, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition'.

Gompel, M. v. & van den Bosch, A. (2013), 'Parameter optimisation for memory-based cross-lingual word-sense disambiguation'.

Gonzales, A. R., Mascarell, L. & Sennrich, R. (2017), Improving word sense disambiguation in neural machine translation with sense embeddings, *in* 'Proceedings of the Second Conference on Machine Translation'.

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep Learning*, MIT Press.

Graham, Y., Baldwin, T., Moffat, A. & Zobel, J. (2017), 'Can machine translation systems be evaluated by the crowd alone', *Natural Language Engineering* .

Graves, A., Fernández, S. & Schmidhuber, J. (2005), Bidirectional lstm networks for improved phoneme classification and recognition, *in* 'International Conference on Artificial Neural Networks', Springer, pp. 799–804.

Graves, A. & Schmidhuber, J. (2005), 'Framewise phoneme classification with bidirectional lstm and other neural network architectures', *In proceedings of Neural Networks* .

Grubinger, M., Clough, P., Müller, H. & Deselaers, T. (2006), The iapr tc-12 benchmark: A new evaluation resource for visual information systems, *in* 'OntoImage 2006 Workshop on Language Resources for Content-based Image Retrieval during LREC 2006 Final Programme'.

Hadiwinoto, C., Ng, H. T. & Gan, W. C. (2019), Improved word sense disambiguation using pre-trained contextualized word representations, *in* 'Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)'.

Harris, Z. (1968), 'Mathematical structures of language', *Interscience Tracts in Pure and Applied mathematics* .

He, K., Zhang, X., Ren, S. & Sun, J. (2015), Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, *in* 'Proceedings of the IEEE international conference on computer vision'.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), 'Deep residual learning for image recognition', *In proceedings of the IEEE conference on computer vision and pattern recognition* .

Helcl, J. & Libovický, J. (2017), Cuni system for wmt17 multimodal translation tasks, *in* 'Second Conference on Machine Translation'.

Helcl, J., Libovickỳ, J. & Variš, D. (2018), 'Cuni system for the wmt18 multimodal translation task', *Proceedings of the Third Conference on Machine Translation* .

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. (2012), 'Improving neural networks by preventing co-adaptation of feature detectors', *Computing Research Repository arXiv:1207.0580* .

Hochreiter, S. (1991), 'Untersuchungen zu dynamischen neuronalen netzen. [master's thesis in german. english translation: Investigations into dynamic neural networks.]', *Institut Fur Informatik, Technische Universitat, Munchen.* .

Hochreiter, S. & Schmidhuber, J. (1997), 'Long short-term memory', *In proceedings of Neural Computation* .

Hokamp, C. & Calixto, I. (2016), Multimodal neural machine translation using minimum risk training, *in* 'First Conference on Machine Translation'.

Huang, P.-Y., Liu, F., Shiang, S.-R., Oh, J. & Dyer, C. (2016), Attention-based multimodal neural machine translation, *in* 'First Conference on Machine Translation'.

Hutchins, J. (1997), 'From first conception to first demonstration: The nascent years of machine translation, 1947–1954. a chronology', *Machine Translation* .

Hutchins, W. & Somers, H. (1992), *An Introduction to Machine Translation*, Academic Press.

Iacobacci, I., Pilehvar, M. T. & Navigli, R. (2016), Embeddings for word sense disambiguation: An evaluation study, *in* 'Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)'.

Ide, N. & Véronis, J. (1998), 'Introduction to the special issue on word sense disambiguation: the state of the art', *Computational linguistics* .

Ilievski, F., Postma, M. & Vossen, P. (2016), Semantic overfitting: what 'world' do we consider when evaluating disambiguation of text?, *in* 'Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers'.

Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G. et al. (2017), 'Google's multilingual neural machine translation system: Enabling zero-shot translation', *Transactions of the Association for Computational Linguistics* .

Kadir, T. & Brady, M. (2001), 'Saliency, scale and image description', *International Journal of Computer Vision* .

Kågebäck, M. & Salomonsson, H. (2016), 'Word sense disambiguation using a bidirectional lstm', *COLING 2016* p. 51.

Karpathy, A. & Fei-Fei, L. (2015), Deep visual-semantic alignments for generating image descriptions, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition'.

Kim, J. & Scott, C. (2009), 'L kernel classification', *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Kingma, D. & Ba, J. (2014), Adam: A method for stochastic optimization, *in* 'Proceedings of International Conference on Learning Representations'.

Koehn, P. (2004), Pharaoh: a beam search decoder for phrase-based statistical machine translation models, *in* 'Conference of the Association for Machine Translation in the Americas', Springer.

Koehn, P. (2005), Europarl: A parallel corpus for statistical machine translation, *in* 'Machine Translation summit', Citeseer.

Koehn, P. (2009), *Statistical Machine Translation*, Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. & Herbst, E. (2007), Moses: Open source toolkit for statistical machine translation, *in* 'Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions', ACL '07, Association for Computational Linguistics.

Koehn, P., Och, F. J. & Marcu, D. (2003), Statistical phrase-based translation, *in* 'Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1', NAACL '03, Association for Computational Linguistics.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* 'Advances in Neural Information Processing Systems'.

Kusner, M., Sun, Y., Kolkin, N. & Weinberger, K. (2015), From word embeddings to document distances, *in* 'International Conference on Machine Learning', pp. 957–966.

Lala, C., Madhyastha, P. S., Scarton, C. & Specia, L. (2018), Sheffield submissions for wmt18 multimodal translation shared task, *in* 'Proceedings of the Third Conference on Machine Translation: Shared Task Papers (WMT 2018)'.

Lala, C., Madhyastha, P. S. & Specia, L. (2019), Grounded word sense translation, *in* 'Proceedings of the Second Workshop on Shortcomings in Vision and Language (NAACL 2019)'.

Lala, C., Madhyastha, P., Wang, J. & Specia, L. (2017), 'Unraveling the contribution of image captioning and neural machine translation for multimodal machine translation', *The Prague Bulletin of Mathematical Linguistics (EAMT 2017)* .

Lala, C. & Specia, L. (2018), Multimodal lexical translation, *in* 'proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)'.

Le, C. A. & Shimazu, A. (2004), High wsd accuracy using naive bayesian classifier with rich features, *in* 'Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation'.

Le, D.-T., Uijlings, J. & Bernardi, R. (2014), TUHOI: Trento universal human object interaction dataset, *in* 'Proceedings of the Third Workshop on Vision and Language', Dublin City University and the Association for Computational Linguistics.

Lee, Y. K. & Ng, H. T. (2002), An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation, *in* 'Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)'.

Lee, Y. K., Ng, H. T. & Chia, T. K. (2004), Supervised word sense disambiguation with support vector machines and multiple knowledge sources, *in* 'Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text'.

Lefever, E. & Hoste, V. (2010), Semeval-2010 task 3: Cross-lingual word sense disambiguation, *in* '5th International Workshop on Semantic Evaluation (SemEval 2010)', Association for Computational Linguistics (ACL).

Lefever, E. & Hoste, V. (2013), Semeval-2013 task 10: Cross-lingual word sense disambiguation, *in* 'Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)'.

Lesk, M. (1986), Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone, *in* 'Proceedings of the 5th Annual International Conference on Systems Documentation'.

Li, X., Lan, W., Dong, J. & Liu, H. (2016), Adding chinese captions to images, *in* 'Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval'.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F. et al. (2020), Oscar: Object-semantics aligned pre-training for vision-language tasks, *in* 'European Conference on Computer Vision', Springer, pp. 121–137.

Libovický, J., Helcl, J., Tlustý, M., Bojar, O. & Pecina, P. (2016), Cuni system for wmt16 automatic post-editing and multimodal translation tasks, *in* 'First Conference on Machine Translation'.

Lin, C.-Y. (2004), Rouge: A package for automatic evaluation of summaries, *in* 'Proceedings of Workshop on Text Summarization Branches Out, Post Conference Workshop of ACL 2004'.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. (2014), Microsoft coco: Common objects in context, *in* 'European Conference on Computer Vision', Springer.

Linden, K., Axelson, E., Drobac, S., Hardwick, S., Kuokkala, J., Niemi, J., Pirinen, T. & Silfverberg, M. (2013), *HFST—a System for Creating NLP Tools*, Communications in Computer and Information Science, Springer-Verlag.

Liu, F., Lu, H. & Neubig, G. (2018), Handling homographs in neural machine translation, *in* 'Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)'.

Loeff, N., Alm, C. O. & Forsyth, D. A. (2006), Discriminating image senses by clustering with multimodal features, *in* 'Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics'.

Loper, E. & Bird, S. (2002), Nltk: The natural language toolkit, *in* 'Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics'.

Luong, T., Pham, H. & Manning, C. D. (2015), Effective approaches to attention-based neural machine translation, *in* 'Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics.

Madhyastha, P. S., Wang, J. & Specia, L. (2019), Vifidel: Evaluating the visual fidelity of image descriptions, *in* 'Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics'.

Madhyastha, P., Wang, J. & Specia, L. (2017), Sheffield multimt: Using object posterior predictions for multimodal machine translation, *in* 'Second Conference on Machine Translation'.

Madhyastha, P., Wang, J. & Specia, L. (2018), 'The role of image representations in vision to language tasks', *Natural Language Engineering* .

Mallery, J. C. (1988), Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers, *in* 'Master's thesis, MIT Political Science Department', Citeseer.

Marvin, R. & Koehn, P. (2018), Exploring word sense disambiguation abilities of neural machine translation systems (non-archival extended abstract), *in* 'Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)'.

Mascarell, L. (2017), Lexical chains meet word embeddings in document-level statistical machine translation, *in* 'Proceedings of the Third Workshop on Discourse in Machine Translation'.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), 'Efficient estimation of word representations in vector space', *arXiv preprint arXiv:1301.3781* .

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013), Distributed representations of words and phrases and their compositionality, *in* 'Advances in Neural Information Processing Systems'.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D. & Miller, K. J. (1990), 'Introduction to wordnet: An on-line lexical database', *International Journal of Lexicography* .

Miller, G. A., Chodorow, M., Landes, S., Leacock, C. & Thomas, R. G. (1994), Using a semantic concordance for sense identification, *in* 'Proceedings of the workshop on Human Language Technology'.

Mooney, R. (1996), Comparative experiments on disambiguation word senses: An illustration of the role of bias in machine learning, *in* 'Proceedings of the Conference on Empirical Methods in Natural Language Processing'.

Navigli, R. (2009), 'Word sense disambiguation: A survey', *ACM Computing Surveys* .

Navigli, R., Jurgens, D. & Vannella, D. (2013), Semeval-2013 task 12: Multilingual word sense disambiguation, *in* 'Second Joint Conference on Lexical and Computational Semantics (* SEM)'.

Navigli, R. & Ponzetto, S. P. (2012), 'Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network', *Artificial Intelligence* .

Ng, A. Y., Jordan, M. I. & Weiss, Y. (2002), On spectral clustering: Analysis and an algorithm, *in* 'Advances in neural information processing systems', pp. 849–856.

Och, F. J. (2002), Statistical Machine Translation: From Single-Word Models to Alignment Templates, PhD thesis, Bibliothek der RWTH Aachen.

Och, F. J. (2003), Minimum error rate training in statistical machine translation, *in* 'Proceedings of the 41st annual meeting of the Association for Computational Linguistics'.

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D. A., Eng, K. et al. (2004), A smorgasbord of features for statistical machine translation, *in* 'Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004', pp. 161–168.

Och, F. J. & Ney, H. (2003), 'A systematic comparison of various statistical alignment models', *Computational Linguistics* .

Ordonez, V., Kulkarni, G. & Berg, T. L. (2011), Im2text: Describing images using 1 million captioned photographs, *in* 'Advances in Neural Information Processing Systems'.

Papineni, K., Roukos, S., Ward, T. & jing Zhu, W. (2002), 'BLEU: A Method for Automatic Evaluation of Machine Translation', *In proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* .

Pearson, K. (1901), 'Liii. on lines and planes of closest fit to systems of points in space', *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* .

Pelevina, M., Arefyev, N., Biemann, C. & Panchenko, A. (2017), 'Making sense of word embeddings', *Proceedings of the 1st Workshop on Representation Learning for NLP* .

Pennington, J., Socher, R. & Manning, C. D. (2014), Glove: Global vectors for word representation, *in* 'Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)'.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. & Zettlemoyer, L. (2018), Deep contextualized word representations, *in* 'Proceedings of NAACL-HLT'.

Petrolito, T. & Bond, F. (2014), A survey of wordnet annotated corpora.

Plank, B., Søgaard, A. & Goldberg, Y. (2016), Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss, *in* 'Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)'.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J. & Lazebnik, S. (2015), Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, *in* 'Proceedings of the IEEE international conference on computer vision'.

Popov, A. (2017), Word sense disambiguation with recurrent neural networks.

Popović, M. (2011), 'Hjerson: An open source tool for automatic error classification of machine translation output', *The Prague Bulletin of Mathematical Linguistics* .

Popović, M. & Ney, H. (2011), 'Towards automatic error analysis of machine translation output', *Computational Linguistics* **37**(4), 657–688.

Porter, M. F. (2001), 'Snowball: A language for stemming algorithms'.

Postma, M., Bevia, R. I. & Vossen, P. (2016), More is not always better: balancing sense distributions for all-words word sense disambiguation, *in* 'Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers'.

Raganato, A., Bovi, C. D. & Navigli, R. (2017), 'Neural sequence learning models for word sense disambiguation', *In proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* .

Raganato, A. & Tiedemann, J. (2018), An analysis of encoder representations in transformer-based machine translation, *in* 'Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP', Association for Computational Linguistics.

Rashtchian, C., Young, P., Hodosh, M. & Hockenmaier, J. (2010), Collecting image annotations using amazon's mechanical turk, *in* 'Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk'.

Ren, S., He, K., Girshick, R. & Sun, J. (2015), Faster r-cnn: Towards real-time object detection with region proposal networks, *in* 'Advances in Neural Information Processing Systems'.

Resnik, P. & Yarowsky, D. (1999), 'Distinguishing systems and distinguishing senses: New evaluation methods for word sense disambiguation', *Natural Language Engineering* .

Reynolds, D. A. (2009), 'Gaussian mixture models', *Encyclopedia of Biometrics* .

Riedl, M. & Biemann, C. (2016), Unsupervised compound splitting with distributional semantics rivals supervised methods, *in* 'Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies'.

Roukos, S., Graff, D. & Melamed, D. (1995), Hansard french/english ldc95t20, Philadelphia: Linguistic Data Consortium.

Rudnick, A., Liu, C. & Gasser, M. (2013), Hltdi: Cl-wsd using markov random fields for semeval-2013 task 10, *in* 'Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)'.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), 'Learning representations by back-propagating errors', *Nature* .

Russakovsky, O., Deng, J., Huang, Z., Berg, A. C. & Fei-Fei, L. (2013), Detecting avocados to zucchinis: what have we done, and where are we going?, *in* 'International Conference on Computer Vision (ICCV)'.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al. (2015), 'Imagenet large scale visual recognition challenge', *International Journal of Computer Vision* .

Saenko, K. & Darrell, T. (2009), Unsupervised learning of visual sense models for polysemous words, *in* 'Advances in Neural Information Processing Systems'.

Scarlini, B., Pasini, T. & Navigli, R. (2020), Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation., *in* 'AAAI'.

Schroff, F., Kalenichenko, D. & Philbin, J. (2015), Facenet: A unified embedding for face recognition and clustering, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition'.

Sennrich, R. & Haddow, B. (2016), Linguistic input features improve neural machine translation, *in* 'Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers'.

Shah, K., Wang, J. & Specia, L. (2016), Shef-multimodal: Grounding machine translation on images, *in* 'First Conference on Machine Translation'.

Shannon, C. E., Weaver, W. et al. (1949), 'A mathematical theory of communication'.

Sharma, P., Ding, N., Goodman, S. & Soricut, R. (2018), Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning, *in* 'Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)'.

Simonyan, K. & Zisserman, A. (2015), Very deep convolutional networks for large-scale image recognition, *in* Y. Bengio & Y. LeCun, eds, '3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings'.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L. & Makhoul, J. (2006), A study of translation edit rate with targeted human annotation, *in* 'Association for Machine Translation in the Americas'.

Specia, L., Arora, R., Barrault, L., Caglayan, O., Duarte, A., Elliott, D., Gella, S., Holzenberger, N., Lala, C., Lee, S. J. et al. (2020), 'Grounded sequence to sequence transduction', *IEEE Journal of Selected Topics in Signal Processing* .

Specia, L., Frank, S., Sima'an, K. & Elliott, D. (2016), A shared task on multimodal machine translation and crosslingual image description, *in* 'Proceedings of the First Conference on Machine Translation'.

Specia, L., Harris, K., Burchardt, A., Turchi, M., Negri, M. & Skadina, I. (2017), Translation quality and productivity: A study on rich morphology languages., *in* 'Machine Translation Summit XVI'.

Specia, L., Sankaran, B. & Nunes, M. d. G. V. (2008), N-best reranking for the efficient integration of word sense disambiguation and statistical machine translation, *in* 'International Conference on Intelligent Text Processing and Computational Linguistics', Springer.

Straková, J., Straka, M. & Hajič, J. (2014), Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition, *in* 'Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations', Association for Computational Linguistics.

Su, H., Deng, J. & Fei-Fei, L. (2012), Crowdsourcing annotations for visual object detection, *in* 'Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence'.

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F. & Dai, J. (2020), Vl-bert: Pre-training of generic visual-linguistic representations, *in* 'International Conference on Learning Representations'.

Sutskever, I., Vinyals, O. & Le, Q. V. (2014), Sequence to sequence learning with neural networks, *in* 'Advances in Neural Information Processing Systems'.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015), Going deeper with convolutions, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition'.

Taigman, Y., Yang, M., Ranzato, M. & Wolf, L. (2014), Deepface: Closing the gap to human-level performance in face verification, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition'.

Tillmann, C. & Ney, H. (2003), 'Word reordering and a dynamic programming beam search algorithm for statistical machine translation', *Computational linguistics* .

Tokowicz, N. & Degani, T. (2010), 'Translation ambiguity: Consequences for learning and processing', *Research on second language processing and parsing* .

Toral, A. & Sánchez-Cartagena, V. M. (2017), A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions, *in* 'Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers', Association for Computational Linguistics.

Unal, M. E., Citamak, B., Yagcioglu, S., Erdem, A., Erdem, E., Cinbis, N. I. & Cakici, R. (2016), Tasviret: A benchmark dataset for automatic turkish description generation from images, *in* '2016 24th Signal Processing and Communication Application Conference (SIU)', IEEE.

Vardaro, J., Schaeffer, M. & Hansen-Schirra, S. (2019), Translation quality and error recognition in professional neural machine translation post-editing, *in* 'Informatics', Multidisciplinary Digital Publishing Institute.

Vasilescu, F., Langlais, P. & Lapalme, G. (2004), Evaluating variants of the lesk approach for disambiguating words., *in* 'proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)'.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), Attention is all you need, *in* 'Advances in Neural Information Processing Systems'.

Vedantam, R., Lawrence Zitnick, C. & Parikh, D. (2015), Cider: Consensus-based image description evaluation, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 4566–4575.

Vickrey, D., Biewald, L., Teyssier, M. & Koller, D. (2005), Word-sense disambiguation for machine translation, *in* 'Proceedings of human language technology conference and conference on empirical methods in natural language processing', pp. 771–778.

Vinyals, O., Toshev, A., Bengio, S. & Erhan, D. (2015), Show and tell: A neural image caption generator, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition'.

Vinyals, O., Toshev, A., Bengio, S. & Erhan, D. (2016), 'Show and tell: Lessons learned from the 2015 mscoco image captioning challenge', *IEEE transactions on pattern analysis and machine intelligence* .

Wang, J., Madhyastha, P. S. & Specia, L. (2018), Object counts! bringing explicit detections back into image captioning, *in* 'Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)', pp. 2180–2193.

Weaver, W. (1949), Translation, *in* 'Machine Translation of Languages: Fourteen Essays', MIT Press.

Webb, G. I. (2010), *Naïve Bayes*, Springer US.

Werbos, P. J. (1990), 'Backpropagation through time: what it does and how to do it', *Proceedings of the IEEE* .

Wiedemann, G., Remus, S., Chawla, A. & Biemann, C. (2019), 'Does bert make any sense? interpretable word sense disambiguation with contextualized embeddings', *arXiv preprint arXiv:1909.10430* .

Willett, P. (2006), 'The porter stemming algorithm: then and now', *Program* .

Wu, Q., Shen, C., Liu, L., Dick, A. & Van Den Hengel, A. (2016), What value do explicit high level concepts have in vision to language problems?, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition'.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M. & Dean, J. (2016), 'Google's neural machine translation system: Bridging the gap between human and machine translation', *CoRR* **abs/1609.08144**.
URL: *http://arxiv.org/abs/1609.08144*

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. & Bengio, Y. (2015), Show, attend and tell: Neural image caption generation with visual attention, *in* 'International Conference on Machine Learning'.

You, Q., Jin, H., Wang, Z., Fang, C. & Luo, J. (2016), Image captioning with semantic attention, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition'.

Young, P., Lai, A., Hodosh, M. & Hockenmaier, J. (2014), 'From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions', *Transactions of the Association for Computational Linguistics* .

Yuan, D., Richardson, J., Doherty, R., Evans, C. & Altendorf, E. (2016), 'Semi-supervised word sense disambiguation with neural models', *arXiv preprint arXiv:1603.07012* .

Zar, J. H. (2014), 'Spearman rank correlation: overview', *Wiley StatsRef: Statistics Reference Online* .

Zeiler, M. D. & Fergus, R. (2013), 'Stochastic pooling for regularization of deep convolutional neural networks', *Computing Research Repository arXiv:1301.3557* .

Zeiler, M. D. & Fergus, R. (2014), Visualizing and understanding convolutional networks, *in* ‘European conference on computer vision’, Springer.

Zhang, X. (2010), *Support Vector Machines*, Springer US, Boston, MA.

Appendices

Appendix A

Mathematics and Statistics

A.1 Bayes' Theorem

Let A and B be two events. Denote $p(A)$ and $p(B)$ as the probabilities of observing events A and B respectively. Let $p(B) \neq 0$. Then the conditional probability $p(A|B)$ of observing event A given that event B has occurred is given by the following formula:

$$p(A|B) = \frac{p(B|A) \cdot p(A)}{p(B)} \tag{A.1}$$

A.2 Product Rule

For random events X and Y, the joint probability $p(X, Y)$ of observing both X and Y together can be computed from the probability $p(Y)$ of observing Y and the conditional probability $p(X|Y)$ of observing event X given Y has already occurred as follows:

$$p(X, Y) = p(Y) \cdot p(X|Y) \tag{A.2}$$

A.3 Chain Rule

For n random events $X_1, X_2, ..., X_n$; the joint probability $p(X_n, X_{n-1}, X_{n-2}, ..., X_1)$ of observing all the n events together can be computed from the joint probability of observing $n-1$ events together and the conditional probability of observing the n^{th} event X_n given the rest of the $n-1$ events have occurred using the product rule (Appendix A.2) as follows:

$$p(X_n, X_{n-1}, X_{n-2}, ..., X_1) = p(X_{n-1}, X_{n-2}, ..., X_1) \cdot p(X_n|X_{n-1}, X_{n-2}, ..., X_1) \tag{A.3}$$

By repeatedly applying the product rule (Appendix A.2) to each joint probability term, we get the chain rule as follows:

$$p(X_n, X_{n-1}, X_{n-2}, ..., X_1) = p(X_1) \cdot \prod_{k=2}^{n} p(X_k|X_{k-1}, X_{k-2}, ..., X_1) \tag{A.4}$$

A.4 Activation Functions

Here are some activation functions commonly used in Neural Networks.

A.4.1 Identity

$$f(x) = I(x) = x \tag{A.5}$$

Its derivative is $f'(x) = 1$. Its range is $(-\infty, +\infty)$.

A.4.2 Logistic / Sigmoid

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{A.6}$$

Its derivative is $f'(x) = f(x)(1 - f(x))$. Its range is $(0, 1)$.

A.4.3 TanH

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{A.7}$$

Its derivative is $f'(x) = 1 - f(x)^2$. Its range is $(-1, 1)$.

A.4.4 Rectified Linear Unit (ReLU)

$$f(x) = \begin{cases} 0 \text{ if } x \leq 0 \\ x \text{ if } x > 0 \end{cases} \tag{A.8}$$

Its derivative is $f'(x) = \begin{cases} 0 \text{ if } x \leq 0 \\ 1 \text{ if } x > 0 \end{cases}$
Its range is $[0, +\infty)$

A.5 Entropy and Perplexity

Let x be a random variable and $p(x)$ be its probability distribution then its entropy $H(p)$ is given by the formula:

$$H(p) = -\sum_x p(x) \log p(x) \tag{A.9}$$

Perplexity $PP(p)$ is the exponentiation of entropy given by the formula:

$$PP(p) = \exp(H(p)) = \exp(-\sum_x p(x) \log p(x)) \tag{A.10}$$

In Natural Language Processing, if w is sequence of n words and $p(w)$ is the language model probability of w then the perplexity formula above (Equation A.11) gets simplified to the following formula:

$$PP(p(w)) = p(w)^{-\frac{1}{n}} \tag{A.11}$$

A.6 Precision, Recall, Accuracy and F-score

Consider a binary classification task where the two classes are *positive* and *negative.* If a classifier correctly classifies a test sample to be positive then it is called a *true positive*, and if it incorrectly classifies a test sample to be positive then it is called a *false positive.* Similarly, if a classifier correctly classified a test sample to be negative then it is called a *true negative*, and it it incorrectly classifies a test sample to be negative then it is called *false negative.* Let tp be the total number of true positives in a test set. Similarly, let fp, tn and fn be the total number of false positives, true negatives and false negatives respectively.

Precision measures how often is the classifier correct when it assigns the positive label.

$$\text{Precision} = \frac{tp}{tp + fp} \tag{A.12}$$

Recall measures how often a positive labeled sample is correctly classified by the classifier.

$$\text{Recall} = \frac{tp}{tp + fn} \tag{A.13}$$

Accuracy measures the total number of times the classifier was correct.

$$\text{Accuracy} = \frac{tp + tn}{tp + fp + tn + fn} \tag{A.14}$$

F-measure is a weighted harmonic mean of precision and recall.

$$\text{F}_\alpha = (1 + \alpha^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\alpha^2 \cdot \text{Precision} + \text{Recall}} \tag{A.15}$$

In practice, F_1-measure is commonly used.

$$\text{F}_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{A.16}$$

A.7 Cosine Similarity

Let $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ be vectors in n-dimensional vector space R^n. Then cosine similarity C between x and y is given as follows:

$$C(x, y) = \frac{x \cdot y}{||x||||y||} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{A.17}$$

A.8 Jaccard Similarity

Let A and B be two sets of points, then Jaccard similarity J between the two sets is simply the intersection of the two sets divided by the union of the two sets a follows:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{A.18}$$

In computer vision, Jaccard similarity is used to measure the similarity between two bounding boxes. Its called ‘Intersection over Union’.

A.9 TF-IDF: Term Frequency - Inverse Document Frequency

Term Frequency - Inverse Document Frequency is a formulation to quantify the importance of a word within a document. Given a word or a term t and a document d in a collection of documents D, we define (1) Term Frequency TF and (2) Inverse Document Frequency IDF as follows:

$$TF(t,d) = \text{Number of times the term } t \text{ appears in the document } d \tag{A.19}$$

$$IDF(t,d) = \frac{|D|}{|\{d \in D \text{ such that } t \in d\}|} \tag{A.20}$$

There can be other formulations of Term Frequency and Inverse Document Frequency. Finally, TFIDF is simply the product of the Term Frequency and Inverse Document Frequency as follows:

$$TFIDF(t,d) = TF(t,d) \cdot IDF(t,d) \tag{A.21}$$

A.10 Pearson's Correlation Coefficient

Given paired data $\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$, the Pearson's Correlation Coefficient ρ_p between x and y is as follows:

$$\rho_p(x,y) = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{A.22}$$

where $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$ and $\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$.

Pearson's Correlation Coefficient computes the linear correlation between two variables x and y which could be, for example, two metrics evaluating various Multimodal Machine Translation systems. It is a number between -1 and +1 where -1 means total negative correlation, +1 means total positive correlation and 0 means there is no correlation between the two variables.

A.11 Spearman's Rank Correlation Coefficient

Given paired data $\{(x_1,y_1),(x_2,y_2),...,(x_n,y_n)\}$, we will first define rank r of data point (x_i,y_i) with respect to a variable x or y, denoted as $r_x(x_i,y_i)$ or $r_y(x_i,y_i)$. For the variable x, we will first arrange the datapoints in the increasing order of their x-coordinate values. Then the first datapoint, say (x_k,y_k) in this ordered list of datapoints gets the rank 1 (In other words, $r_x(x_k,y_k) = 1$). The next datapoint in that ordered list of data point gets the rank 2, and so on. Similarly to get the ranks r with respect to y, we will arrange the datapoints in the increasing order of their y-coordinate values. Then $r_y(x_k,y_k)$ is the rank/position of the datapoint (x_k,y_k) in this newly ordered list of datapoints.

Finally, Spearman's Rank Correlation Coefficient ρ_s of x and y is simple the Pearson's Correlation Coefficient (Appendix A.10) of the rank values r_x and r_y as follows:

$$\rho_s(x,y) = \rho_p(r_x, r_y) = \frac{\sum_{i=1}^{n}(r_x(x_i,y_i) - \bar{r_x})(r_y(x_i,y_i) - \bar{r_y})}{\sqrt{\sum_{i=1}^{n}(r_x(x_i,y_i) - \bar{r_x})^2}\sqrt{\sum_{i=1}^{n}(r_y(x_i,y_i) - \bar{r_y})^2}} \tag{A.23}$$

where $\bar{r_x} = \bar{r_y} = \frac{n+1}{2}$.

Like Pearson's Correlation coefficient, Spearman's Rank Correlation coefficient is also a number between -1 and +1, where -1 means total negative correlation, +1 means total positive correlation and 0 means there is no correlation between the two kinds of ranking r_x and r_y as defined by the variables x and y respectively.

A.12 Softmax

Consider a vector $x = (x_1, x_2, ..., x_n)$, then we define Softmax(x) as follows:

$$\text{Softmax}(x) = (\frac{\exp^{x_1}}{\sum_{j=1}^{n}\exp^{x_j}}, \frac{\exp^{x_2}}{\sum_{j=1}^{n}\exp^{x_j}}, ..., \frac{\exp^{x_n}}{\sum_{j=1}^{n}\exp^{x_j}}) \tag{A.24}$$

Softmax is commonly used in the output layer of a Neural Network for multiclass classification. It can be thought of as representing a posterior probability distribution over the classes.

A.13 Distance Metrics

Let $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$ be two vectors. Then some of the different distance metrics d are,

A.13.1 Cosine Distance

Then Cosine Distance is simply 1 minus cosine similarity (see Appendix A.7),

$$d(x,y) = 1 - \frac{x \cdot y}{||x||||y||} = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{A.25}$$

A.13.2 Euclidean Distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{A.26}$$

A.13.3 Manhattan Distance

$$d(x,y) = \sum_{i=1}^{n}|x_i - y_i| \tag{A.27}$$

A.13.4 Chebyshev Distance

$$d(x, y) = \max_{i=1}^{n} |x_i - y_i| \tag{A.28}$$

A.13.5 Minkowski Distance

$$d(x, y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}} \tag{A.29}$$

Where p can be any real number. In our experiments, we explored $p \in \{3, 4\}$. $p = 1$ refers to Manhattan Distance, $p = 2$ is Euclidean Distance and $p = \infty$ corresponds to Chebyshev Distance.

A.14 Cross Entropy Loss

Let x be a random variable, $p(x)$ be its true probability distribution and $q(x)$ be an estimated probability distributions, then cross entropy loss L is similar to equation A.9 as follows,

$$L(p, q) = -\sum_{x} p(x) \log q(x) \tag{A.30}$$

In practice, for a neural network, for a given input x let $\hat{y}$ be it's ground truth label represented as a one-hot vector with 1 in some i^{th} position corresponding to the true label and 0 elsewhere. This can be considered as the true posterior probability distribution of the example. A model's posterior distribution over classes $p(y|x)$ (usually obtained by Softmax) can be thought of as an estimated probability distribution over classes/labels. Therefore, cross entropy loss L in this context is the dot product as follows,

$$L(\hat{y}, p(y|x)) = -\hat{y} \cdot \log p(y|x) \tag{A.31}$$

Appendix B

Results

B.1 Sense-preserving Lexical Translation Accuracy measure of systems submitted to the Multimodal Machine Translation shared task 2017

System	**LTA ↑**	Meteor ↑	Human ↑
NICT_1_NMTrerank_C	**75.5**	53.9	70.3
LIUMCVC_NMT_C	**74.7**	53.8	65.1
LIUMCVC_MNMT_C	**73.8**	54.0	77.8
DCU-ADAPT_MultiMT_C	**71.5**	50.5	68.1
UvA-TiCC_IMAGINATION_U	**70.8**	53.5	74.1
UvA-TiCC_IMAGINATION_C	**70.8**	51.2	59.7
CUNI_NeuralMonkeyTextualMT_U	**70.0**	51.0	68.1
CUNI_NeuralMonkeyMultimodalMT_U	**69.3**	50.2	60.6
OREGONSTATE_2NeuralTranslation_C	**68.6**	50.6	54.4
CUNI_NeuralMonkeyTextualMT_C	**67.7**	49.2	54.2
CUNI_NeuralMonkeyMultimodalMT_C	**65.0**	47.1	55.9
OREGONSTATE_1NeuralTranslation_C	**64.8**	48.9	53.3
SHEF_ShefClassProj_C	**60.7**	43.4	49.4
SHEF_ShefClassInitDec_C	**60.5**	44.5	46.6
AFRL-OHIOSTATE-MULTIMODAL_U	**23.1**	20.2	36.6

Table B.1: *Performance of systems submitted to the Multimodal Machine Translation shared task 2017. Submitted systems translated from English to German. Test 2017 Flickr dataset was used to test the systems. LTA stands for sense-preserving Lexical Translation Accuracy. All metrics measured in percentages (%) between 0 and 100.*

System	**LTA** ↑	Meteor ↑	Human ↑
NICT_1_NMTrerank_C	**82.5**	72.0	79.4
LIUMCVC_NMT_C	**81.3**	70.1	60.5
LIUMCVC_MNMT_C	**81.2**	72.1	71.2
DCU-ADAPT_MultiMT_C	**81.0**	70.1	74.1
OREGONSTATE_2NeuralTranslation_C	**78.7**	68.3	65.4
OREGONSTATE_1NeuralTranslation_C	**75.8**	67.2	60.8
CUNI_NeuralMonkeyTextualMT_C	**75.6**	67.0	61.9
CUNI_NeuralMonkeyMultimodalMT_C	**75.0**	67.2	74.2
SHEF_ShefClassInitDec_C	**73.7**	62.8	54.7
SHEF_ShefClassProj_C	**72.4**	61.5	54.0

Table B.2: *Performance of systems submitted to the Multimodal Machine Translation shared task 2017. Submitted systems translated from English to French. Test 2017 Flickr dataset was used to test the systems. LTA stands for sense-preserving Lexical Translation Accuracy. All metrics measured in percentages (%) between 0 and 100.*

System	**LTA** ↑	Meteor ↑
DCU-ADAPT_MultiMT_C	**68.5**	46.8
LIUMCVC_NMT_C	**68.2**	48.9
NICT_1_NMTrerank_C	**67.2**	48.5
UvA-TiCC_IMAGINATION_C	**67.2**	45.8
LIUMCVC_MNMT_C	**66.4**	48.8
CUNI_NeuralMonkeyMultimodalMT_U	**65.4**	45.6
UvA-TiCC_IMAGINATION_U	**64.3**	48.1
CUNI_NeuralMonkeyTextualMT_U	**63.8**	46.0
OREGONSTATE_1NeuralTranslation_C	**63.0**	46.5
OREGONSTATE_2NeuralTranslation_C	**63.0**	45.7
CUNI_NeuralMonkeyTextualMT_C	**63.0**	43.8
CUNI_NeuralMonkeyMultimodalMT_C	**57.7**	42.7
SHEF_ShefClassProj_C	**55.6**	40.0
SHEF_ShefClassInitDec_C	**54.3**	40.7

Table B.3: *Performance of systems submitted to the Multimodal Machine Translation shared task 2017. Submitted systems translated from English to German. Test 2017 MSCOCO dataset was used to test the systems. LTA stands for sense-preserving Lexical Translation Accuracy. All metrics measured in percentages (%) between 0 and 100.*

System	**LTA** ↑	Meteor ↑
LIUMCVC_MNMT_C	**77.6**	65.9
NICT_1_NMTrerank_C	**77.6**	65.6
DCU-ADAPT_MultiMT_C	**76.4**	64.1
LIUMCVC_NMT_C	**75.3**	63.4
OREGONSTATE_2NeuralTranslation_C	**74.8**	63.8
CUNI_NeuralMonkeyTextualMT_C	**74.8**	62.5
CUNI_NeuralMonkeyMultimodalMT_C	**74.8**	62.5
OREGONSTATE_1NeuralTranslation_C	**70.8**	61.6
SHEF_ShefClassProj_C	**68.9**	57.0
SHEF_ShefClassInitDec_C	**68.5**	57.3

Table B.4: *Performance of systems submitted to the Multimodal Machine Translation shared task 2017. Submitted systems translated from English to French. Test 2017 MSCOCO dataset was used to test the systems. LTA stands for sense-preserving Lexical Translation Accuracy. All metrics measured in percentages (%) between 0 and 100.*

B.2 Sense-preserving Lexical Translation Accuracy measure of systems submitted to the Multimodal Machine Translation shared task 2017 with Skewness Ratio > 1.2

System	**LTA$_{1.2}$** ↑	Meteor ↑	Human ↑
NICT_1_NMTrerank_C	**69.1**	53.9	70.3
LIUMCVC_NMT_C	**69.1**	53.8	65.1
LIUMCVC_MNMT_C	**68.1**	54.0	77.8
DCU-ADAPT_MultiMT_C	**65.8**	50.5	68.1
UvA-TiCC_IMAGINATION_U	**65.4**	53.5	74.1
CUNI_NeuralMonkeyTextualMT_U	**64.0**	51.0	68.1
UvA-TiCC_IMAGINATION_C	**63.0**	51.2	59.7
OREGONSTATE_2NeuralTranslation_C	**62.2**	50.6	54.4
CUNI_NeuralMonkeyMultimodalMT_U	**61.6**	50.2	60.6
CUNI_NeuralMonkeyTextualMT_C	**61.3**	49.2	54.2
OREGONSTATE_1NeuralTranslation_C	**59.7**	48.9	53.3
CUNI_NeuralMonkeyMultimodalMT_C	**58.7**	47.1	55.9
SHEF_ShefClassProj_C	**56.2**	43.4	49.4
SHEF_ShefClassInitDec_C	**53.8**	44.5	46.6
AFRL-OHIOSTATE-MULTIMODAL_U	**18.4**	20.2	36.6

Table B.5: *Performance of systems submitted to the Multimodal Machine Translation shared task 2017. Submitted systems translated from English to German. Test 2017 Flickr dataset, consisting only those ambiguous words which have a Skewness Ratio > 1.2, was used to test the systems. LTA$_{1.2}$ stands for sense-preserving Lexical Translation Accuracy with Skewness Ratio > 1.2. All metrics measured in percentages (%) between 0 and 100.*

System	**LTA$_{1.2}$** ↑	Meteor ↑	Human ↑
NICT_1_NMTrerank_C	**71.4**	72.0	79.4
LIUMCVC_MNMT_C	**71.2**	72.1	71.2
DCU-ADAPT_MultiMT_C	**69.6**	70.1	74.1
LIUMCVC_NMT_C	**69.1**	70.1	60.5
OREGONSTATE_2NeuralTranslation_C	**65.7**	68.3	65.4
OREGONSTATE_1NeuralTranslation_C	**63.4**	67.2	60.8
CUNI_NeuralMonkeyTextualMT_C	**60.8**	67.0	61.9
SHEF_ShefClassInitDec_C	**60.0**	62.8	54.7
CUNI_NeuralMonkeyMultimodalMT_C	**59.5**	67.2	74.2
SHEF_ShefClassProj_C	**58.4**	61.5	54.0

Table B.6: *Performance of systems submitted to the Multimodal Machine Translation shared task 2017. Submitted systems translated from English to French. Test 2017 Flickr dataset, consisting only those ambiguous words which have a Skewness Ratio > 1.2, was used to test the systems. $LTA_{1.2}$ stands for sense-preserving Lexical Translation Accuracy with Skewness Ratio > 1.2. All metrics measured in percentages (%) between 0 and 100.*

System	**LTA$_{1.2}$** ↑	Meteor ↑
DCU-ADAPT_MultiMT_C	**65.6**	46.8
LIUMCVC_NMT_C	**64.2**	48.9
UvA-TiCC_IMAGINATION_C	**63.4**	45.8
NICT_1_NMTrerank_C	**60.9**	48.5
UvA-TiCC_IMAGINATION_U	**60.9**	48.1
LIUMCVC_MNMT_C	**59.5**	48.8
OREGONSTATE_2NeuralTranslation_C	**59.1**	45.7
CUNI_NeuralMonkeyMultimodalMT_U	**59.1**	45.6
CUNI_NeuralMonkeyTextualMT_U	**57.7**	46.0
OREGONSTATE_1NeuralTranslation_C	**57.0**	46.5
CUNI_NeuralMonkeyTextualMT_C	**56.6**	43.8
CUNI_NeuralMonkeyMultimodalMT_C	**51.6**	42.7
SHEF_ShefClassInitDec_C	**50.5**	40.7
SHEF_ShefClassProj_C	**49.1**	40.0

Table B.7: *Performance of systems submitted to the Multimodal Machine Translation shared task 2017. Submitted systems translated from English to German. Test 2017 MSCOCO dataset, consisting only those ambiguous words which have a Skewness Ratio > 1.2, was used to test the systems. $LTA_{1.2}$ stands for sense-preserving Lexical Translation Accuracy with Skewness Ratio > 1.2. All metrics measured in percentages (%) between 0 and 100.*

System	**LTA$_{1.2}$ ↑**	Meteor ↑
LIUMCVC_MNMT_C	**66.8**	65.9
NICT_1_NMTrerank_C	**65.8**	65.6
OREGONSTATE_2NeuralTranslation_C	**64.8**	63.8
DCU-ADAPT_MultiMT_C	**64.3**	64.1
LIUMCVC_NMT_C	**63.3**	63.4
CUNI_NeuralMonkeyTextualMT_C	**61.8**	62.5
CUNI_NeuralMonkeyMultimodalMT_C	**61.8**	62.5
OREGONSTATE_1NeuralTranslation_C	**58.3**	61.6
SHEF_ShefClassProj_C	**55.8**	57.0
SHEF_ShefClassInitDec_C	**53.8**	57.3

Table B.8: *Performance of systems submitted to the Multimodal Machine Translation shared task 2017. Submitted systems translated from English to French. Test 2017 MSCOCO dataset, consisting only those ambiguous words which have a Skewness Ratio > 1.2, was used to test the systems. LTA$_{1.2}$ stands for sense-preserving Lexical Translation Accuracy with Skewness Ratio > 1.2. All metrics measured in percentages (%) between 0 and 100.*

B.3 Sense-preserving Lexical Translation Accuracy measure of systems submitted to the Multimodal Machine Translation shared task 2018.

System	BLEU ↑	Meteor ↑	TER ↓	**LTA ↑**	Human ↑
MeMAD_1_FLICKR_DE_MeMAD-OpenNMT-mmod_U	38.5	56.6	44.5	**47.5**	87.2
CUNI_1_FLICKR_DE_NeuralMonkeyTextual_U	32.5	52.3	50.8	**46.4**	-
CUNI_1_FLICKR_DE_NeuralMonkeyImagination_U	32.2	51.7	51.7	**47.2**	73.8
UMONS_1_FLICKR_DE_DeepGru_C	31.1	51.6	53.4	**48.0**	72.1
LIUMCVC_1_FLICKR_DE_NMTEnsemble_C	31.1	51.5	52.6	**46.7**	72.5
LIUMCVC_1_FLICKR_DE_MNMTEnsemble_C	31.4	51.4	52.1	**45.8**	71.6
OSU-BD_1_FLICKR_DE_RLNMT_C	32.3	50.9	49.9	**45.3**	71.1
OSU-BD_1_FLICKR_DE_RLMIX_C	32.0	50.7	49.6	**46.1**	-
SHEF_1_DE_LT_C	30.4	50.7	53.0	**48.0**	-
SHEF_1_DE_MLT_C	30.4	50.7	53.0	**48.3**	73.5
SHEF1_1_DE_ENMT_C	30.8	50.7	52.4	**44.4**	-
SHEF1_1_DE_MFS_C	30.3	50.7	53.1	**48.3**	72.6
LIUMCVC_1_FLICKR_DE_MNMTSingle_C	28.8	49.9	55.6	**45.3**	-
LIUMCVC_1_FLICKR_DE_NMTSingle_C	29.5	49.9	54.3	**47.8**	-
Baseline	27.6	47.4	55.2	**45.3**	67.4
AFRL-OHIO-STATE_1_FLICKR_DE_4COMBO_U	24.3	45.4	58.6	**46.1**	68.6
AFRL-OHIO-STATE_1_FLICKR_DE_2IMPROVE_U	10.0	25.4	79.0	**25.4**	-

Table B.9: *Performance of systems submitted to the Multimodal Machine Translation shared task 2018. Submitted systems translated from English to German. Test 2018 dataset was used to test the systems. LTA stands for sense-preserving Lexical Translation Accuracy. All metrics measured in percentages (%) between 0 and 100.*

System	BLEU ↑	Meteor ↑	TER ↓	**LTA** ↑	Human ↑
MeMAD_1_FLICKR_FR_MeMAD-OpenNMT-mmod_U	44.1	64.3	36.9	**73.1**	86.8
CUNI_1_FLICKR_FR_NeuralMonkeyTextual_U	40.6	61.0	40.7	**68.4**	-
CUNI_1_FLICKR_FR_NeuralMonkeyImagination_U	40.4	60.7	40.7	**69.3**	78.5
UMONS_1_FLICKR_FR_DeepGru_C	39.2	60.0	41.8	**68.8**	77.3
LIUMCVC_1_FLICKR_FR_MNMTEnsemble_C	39.5	59.9	41.7	**68.5**	73.0
LIUMCVC_1_FLICKR_FR_NMTEnsemble_C	39.1	59.8	41.9	**68.4**	74.9
SHEF_1_FR_LT_C	38.8	59.8	41.5	**69.6**	-
SHEF_1_FR_MLT_C	38.9	59.8	41.5	**69.9**	74.5
SHEF1_1_FR_ENMT_C	38.9	59.8	41.2	**67.9**	-
SHEF1_1_FR_MFS_C	38.8	59.7	41.6	**67.6**	74.9
OSU-BD_1_FLICKR_FR_RLNMT_C	39.0	59.5	41.2	**68.9**	74.4
OSU-BD_1_FLICKR_FR_RLMIX_C	38.6	59.3	41.5	**67.7**	-
LIUMCVC_1_FLICKR_FR_MNMTSingle_C	37.9	58.5	43.4	**67.8**	-
LIUMCVC_1_FLICKR_FR_NMTSingle_C	37.6	58.4	43.2	**67.1**	-
Baseline	36.3	56.9	54.3	**66.3**	66.0

Table B.10: *Performance of systems submitted to the Multimodal Machine Translation shared task 2018. Submitted systems translated from English to French. Test 2018 dataset was used to test the systems. LTA stands for sense-preserving Lexical Translation Accuracy. All metrics measured in percentages (%) 0 to 100.*

System	BLEU ↑	Meteor ↑	TER ↓	**LTA** ↑	Human ↑
CUNI_1_FLICKR_CS_NeuralMonkeyImagination_U	31.8	30.6	48.2	**70.0**	70.2
OSU-BD_1_FLICKR_CS_RLMIX_C	30.1	29.7	51.2	**54.3**	-
OSU-BD_1_FLICKR_CS_RLNMT_C	30.2	29.5	50.7	**60.7**	59.1
SHEF1_1_CS_ENMT_C	29.0	29.4	51.1	**71.4**	-
SHEF1_1_CS_MFS_C	27.8	29.2	52.4	**73.6**	60.6
SHEF_1_CS_LT_C	28.3	29.1	51.7	**72.1**	-
SHEF_1_CS_MLT_C	28.2	29.1	51.7	**71.4**	62.4
Baseline	26.5	27.7	54.4	**62.1**	57.8
CUNI_1_FLICKR_CS_NeuralMonkeyTextual_U	26.8	27.1	55.2	**52.1**	-

Table B.11: *Performance of systems submitted to the Multimodal Machine Translation shared task 2018. Submitted systems translated from English to Czech. Test 2018 dataset was used to test the systems. LTA stands for sense-preserving Lexical Translation Accuracy. All metrics measured in percentages (%) 0 to 100.*

System	BLEU ↑	Meteor ↑	TER ↓	**LTA** ↑	Human ↑
OSU-BD_1b_CS_RLMIX_C	26.4	28.2	52.7	**55.8**	-
OSU-BD_1b_CS_RLNMT_C (P)	26.4	28.0	52.1	**61.5**	62.1
SHEF_1b_CS_CON_C	24.7	27.6	52.1	**61.5**	-
SHEF_1b_CS_MLTC_C (P)	24.5	27.5	52.5	**61.5**	63.3
SHEF1_1b_CS_ARNN_C (P)	25.2	27.5	53.9	**51.9**	61.8
SHEF1_1b_CS_ARF_C	24.1	27.1	54.6	**51.9**	-
Baseline	23.6	26.8	54.1	**53.9**	59.4

Table B.12: *Performance of systems submitted to the Multimodal Machine Translation shared task 2018. Submitted systems translated from English, German and French (combined) to Czech. Test 2018 dataset was used to test the systems. LTA stands for sense-preserving Lexical Translation Accuracy. All metrics measured in percentages (%) between 0 and 100.*