Transfer Learning for Conceptual Metaphor Generation

Bachelor thesis by Nils Peer Beck
Date of submission: Wednesday 2nd June, 2021

1. Review: Prof. Dr. Iryna Gurevych
2. Review: Kevin Stowe, PhD
Darmstadt

TECHNISCHE UNIVERSITÄT DARMSTADT

UBIQUITOUS KNOWLEDGE PROCESSING

Computer Science Department
Ubiquitous Knowledge Processing Lab

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Nils Peer Beck, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 2. Juni 2021

Nils Peer Beck

Abstract

Man kann Metaphern als linguistische Phänomene verstehen, bei denen sich Konzepte aus einer Domäne – z.B. Geld – so verhalten, wie es Konzepte aus einer anderen Domäne – z. B. Flüssigkeit – normalerweise tun. “Ihr Konto wurde *eingefroren*” ist im Sinne von “Conceptual Metaphor Theory” ein Beispiel für einen metaphorischen Ausdruck. In dieser Arbeit befassen wir uns mit dem maschinellen Erzeugen von Metaphern, bei dem ein Eingabetext in einen metaphorischen Ausgabetext umgeformt wird.

Zunächst erstellen wir einen neuartigen Datensatz, bestehend aus mehr als 300.000 metaphorischen Satzpaaren. Dieser Datensatz baut auf zwei lexikalischen Datenbanken der englischen Sprache auf: MetaNet und FrameNet. Anschließend trainieren wir T5, ein umfangreiches vortrainiertes englisches Sprachmodell, in zwei verschiedenen Versuchsaufbauten zur maschinellen Erzeugung von Metaphern: Zum einen trainieren wir ein Sprachmodell zur *freien* Erzeugung von Metaphern, d.h., dass wir vom Sprachmodell erwarten, dass es implizit versteht, wie es den Eingabetext umformen soll. Zum anderen trainieren wir die *kontrollierte* Erzeugung von Metaphern, bei der wir dem Sprachmodell explizit mitteilen, welche Art von Metapher es erzeugen soll.

Wir vergleichen unsere beiden Sprachodelle mit verwandten Arbeiten auf dem Gebiet der maschinellen Erzeugung von Metaphern. Wir berichten über vielversprechende Ergebnisse in beiden Versuchsaufbauten. Zwar nimmt kein Modell entlang aller Evaluierungskriterien den ersten Platz ein, doch unsere Modelle erzeugen besonders gut neuartige Metaphern. Zudem erzeugt unser freies Modell Sätze mit besserem Textfluss, die einander semantisch ähnlicher sind, während das kontrollierte Modell metaphorischere Sätze erzeugt. Es bestehen also je nach Modell verschieden Anwendungsszenarien. Unseren Code, den neuartigen Datensatz, sowie unsere trainierten Sprachmodelle stellen wir für weitere Forschung öffentlich zur Verfügung.

Abstract

Metaphor can be understood as a linguistic phenomenon in which concepts from one domain – for instance, Money – behave the way concepts from a different domain – for instance, Liquid – usually do. “She *drained* her bank account” constitutes an example of a metaphorical expression, according to Conceptual Metaphor Theory. In this thesis, we address the task of metaphor generation, i.e., paraphrasing an input text into a metaphorical output text.

First, we build a novel data set of more than 300,000 metaphorical sentence pairs, anchored in MetaNet and FrameNet, two lexical databases. Then, we fine-tune T5, a large pre-trained English language model, on metaphor generation in two different set-ups: *Free* metaphor generation, where the language model is expected to implicitly understand how to transform the input text, and *controlled* metaphor generation, where the language model is explicitly told what kind of metaphor to generate.

We compare our fine-tuned models to related work in the field of metaphor generation, reporting promising results. While no clear favorite emerges along all evaluation criteria, our models perform particularly well on unseen metaphors. We also show that our free generation model produces more fluent and semantically similar text, while our controlled model produces more metaphorical text, suggesting differing use cases for both models. We make our code, our data set, and our fine-tuned models publicly available for further research.

Contents

1 Introduction

Metaphors form an integral part of human language. Albeit at times considered a figure of speech reserved exclusively for the poetic, artistic, or extravagant occasion, metaphors should rather be understood as a broader, omnipresent phenomenon. They ease humans' understanding of new, distant or abstract ideas, by making these ideas tangible, intuitive, and easier to grasp. If understood in a structural way, metaphors can be seen as depicting one concept – often new, distant, or abstract – in terms of another one, often more familiar [LJ03, Lak93].

"The economy went belly-up".

"The stocks' value peaked."

"The war on covid-19 has only just begun".

Each of these sentences illustrates how a more familiar, tangible concept or experience, e.g., that of rolling your body over, serves to understand a more abstract one, such as sudden changes in the economy, causing people to lose jobs, change their habits, or reconsider previously held beliefs. Much like a human body after having rolled over, the economic system finds itself in a very different state.

Progress in various areas of computer science over the last years – among them machine learning – has powered the computational processing of human language in new and often very successful ways. In this thesis, we focus on a particular field of research called "natural language generation" (NLG), which attempts to translate a computer's internal representation of information into human language [GK18]. Taking into account the role of metaphor in language, we configure computers to generate metaphorical language.

Generating metaphors computationally is relevant both for academia and practitioners. On the side of academia, it would be interesting to see how well conceptual metaphor theory serves to produce metaphorical language, whether we are able to generalize it and can thus create unseen metaphors. We are also interested in finding out whether we can explicitly create novel metaphors that are semantically suitable for a given context, by using control codes.

On a more practical, potentially industry-oriented side, metaphors may also be beneficial when understood as a language *style*, e.g., in making text more comprehensive, compact, engaging, or thought-provoking. A system that is designed specifically to generate metaphorical language might be seen as improving text quality overall.

The remaining chapters of this thesis broadly follow a pipelined approach, with data and ideas from later chapters building on earlier ones, ending up in two computational language models that are able to generate metaphors.

In **chapter 2** we elaborate on previous work in the field of metaphor generation, focusing on approaches powered by deep learning advances, as well as approaches that are informed by a linguistic theory of metaphor. We contrast the previous work with our methodology, outlining differences and similarities.

In **chapter 3**, we introduce the linguistic backbone of this thesis, conceptual metaphor theory (CMT). We explain why metaphors occur and what they mean in a systematic way. Conceptual metaphor theory comprises a promising basis for further computational work. It also provides us with domain names, which we can use as meaningful control codes in *controlled* metaphor generation.

In **chapter 4** we present MetaNet, a lexical database for English, grounded in conceptual metaphor theory. We then introduce ways of mapping MetaNet's structure and elements onto FrameNet, a more extensive lexical resource for English.

In **chapter 5** we offer some thoughts on how metaphor generation can be captured in a computational task description and introduce two definitions for the purposes of this thesis.

In **chapter 6**, we create a novel set of sentence pairs, i.e., tuples consisting of a "standard" input sentence and a "metaphorical" output sentence. We compute these pairs by extracting "standard" sentences from lexical resources (cf. §4)and then replacing individual words within these input sentences with words from a different domain.

In **chapter 7**, we fine-tune a large pretrained language model on the sentence pairs that we created, reaching the end of our language processing pipeline. We do so in a *free* manner – i.e. with just the sentence mappings as training data – as well as in a *controlled* manner, i.e., including control codes for the so-called source domain, which the model is supposed to evoke.

Finally, in **chapter 8**, we evaluate the performance of our fine-tuned language model. We also elaborate on the difficulty of finding evaluation metrics and methods for metaphor generation and other language generation tasks.

In **chapter 9**, we discuss the limitations of this work.

In **chapter 10**, we propose future avenues of research in metaphor generation.

In **chapter 11**, we conclude this thesis.

We publish our code, data and models in publicly available repositories, allowing for all of our results to be reproduced.

2 Related work

While metaphor can be examined from the perspective of cognitive science, linguistics, and many other disciplines, we focus on the computational generation of metaphors from the perspective of computational linguistics, leveraging advances in language modeling that build on deep learning, specifically on the transformer architecture for language models [VSP+17].

We distinguish the task of metaphor generation, where metaphoric mappings are activated in an input sentence, from the tasks of metaphor detection and metaphor interpretation (finding literal counterparts for metaphorical words in a sentence), which have been studied in more depth than metaphor generation.

2.1 Deep Learning & Transformer language models

Chakrabarty et al. (2021) proposed the metaphor generation scheme MERMAID, which in many ways aligns with our approach [CZMP21]. They use a corpus of poetic text, which is assumed to be metaphorical, and then replace metaphorical words with their literal counterparts, yielding a parallel corpus of literal and metaphorical sentence pairs. They then use this novel corpus for transfer learning, fine-tuning the English transformer language model BART [LLG+20] on predicting metaphorical word options for masked words in the input text.

Stowe et al. (2021) follow suit, building a parallel corpus of 248,000 sentence pairs, by walking metaphorical sentences back to equivalent literal counterparts [SCP+21]. Informed by conceptual metaphor theory, they build their parallel training data by leveraging ConceptNet [SCH17] in the search for literal verb candidates to fill the slot of an assumed metaphorical verb. They also tag sentences with corresponding FrameNet frames and use those frame names as control codes for metaphor generation. Finally, they fine-tune BART [LLG+20], a transformer language model for English, with the newly created training data for controlled metaphor generation. While very similar to our approach, the authors do not incorporate MetaNet resources and limit themselves to replacing just verbs. Moreover, they only address controlled metaphor generation, as opposed to free generation.

In an earlier attempt, Stowe et al. (2020) propose MetMask, another scheme to fine-tune an English transformer language model on the task of metaphoric paraphrase generation, i.e., predicting a metaphoric word to fill the slot of a masked word in a given input sentence [SRG20]. They create a corpus of more than 12,000 metaphoric text pairs, along with even more literal pairs.

Yu and Wan (2019) tackle metaphor generation in an unsupervised machine learning set-up, doing away with the need for labelled sentence pairs [YW19]. Given an English verb, they search through a reference corpus, namely WordNet [Mil95], for words that occur in a similar context, but differ semantically. They assume such words to be metaphorical and create a corpus of more than 300,000 metaphorical sentences.

Since their metaphor generation approach relies entirely on WordNet, it is strictly limited to the metaphorical words listed in this database.

Mao et al. (2018) identify and interpret metaphors, also building a corpus of literal and metaphorical sentence pairs [MLG18]. They apply unsupervised machine learning methods, representing words as word embeddings, and replacing infrequent metaphoric words with more frequently occurring counterparts, assuming them to be literal. By stripping text of metaphorical words, they improve machine translation accuracy by up to 11%.

Just like the approaches listed above, we understand metaphor generation as a task in which we paraphrase an input sentence into a more metaphorical output sentence, evoking concepts from what conceptual metaphor theory refers to as a source domain. We also build a parallel corpus of English sentence pairs that is then used for transfer learning on T5, a large transformer language model. In contrast to the above, we resort to knowledge bases that are directly applicable to Conceptual Metaphor Theory, namely MetaNet [DHS15] and FrameNet [Bak15] for constructing a training corpus. Also, in contrast to [SRG20], we only use mask filling while compiling a training corpus, thus allowing for convenient metaphor generation without masks at the time of deploying our fine-tuned language model. Finally, we experiment with control codes for the source domain that is to be evoked, allowing for a comparison of controlled and free metaphor generation.

2.2 Approaches outside of deep learning

Before the advances of large transformer language models, metaphor generation relied more heavily on knowledge bases which can be analyzed statistically without applying computationally demanding deep learning methods. Metaphors were generated as instances of the patterns "A is B" or "A is like B" (cf. [ASN06, TN10, HCC+07]). Ovchinnikova et al. (2014) treat metaphor generation rather like an information retrieval task, searching Russian and English text corpora for metaphorical expressions belonging to a given conceptual metaphor [OZWI14]. Gero and Chilton (2019) implement a human-in-the-loop system that generates some candidates for a metaphorical expression, ranking them based on their GloVe word embeddings [PSM14] [GC19]. A human, interested in improving her creative writing skills, then picks one of the suggestions.

2.3 Early work

Interestingly, metaphor generation was already envisioned at a time when computer scientists lacked the resources and tools to implement it. Jones (1992) draws heavily from conceptual metaphor theory and proposes the task that we refer to as "Controlled Metaphor Generation", where an English input sentence and a control code is given, specifying the "goal that the metaphor is to achieve" (cf. §4, [Jon92]), and a metaphoric output sentence is generated. Jones envisions an algorithm that searches through a hierarchy of metaphor domains, connected by "is-a" relations. He acknowledges that reasoning about the right candidate domain to pick when generating a metaphor would be a hard task to solve. He expects such a metaphor generation algorithm to ease readers' understanding of text, as well as to achieve brevity, conceptual fit, focus, or perspective. Our approach aligns very much with Jones' early ideas, with the exact structured metaphor knowledge bases he calls for – MetaNet and FrameNet – now available for computational search.

Steinhardt (1994) implements NETMET, a computer program that guides humans through the steps of metaphor generation [Ste94]. The author lacks a knowledge base for semantic field theory (closely related to

Conceptual Metaphor Theory), and therefore requires the user to input a set of propositions manually. These propositions are then grouped into semantic fields (referred to as domains in CMT), partly via a clustering algorithm, but mostly via human annotations. When generating a metaphor, a user is guided through the steps of (1) identifying a cue or reference word, (2) attributing it to a target domain, (3) finding an applicable source domain based on the propositions, and finally (4) replacing it with a suitable word from that source domain. Steinhardt's approach is similar to ours, yet he finds himself unable to automate the process of metaphor generation, lacking a reference knowledge base as well as clustering and ranking algorithms that allow him to determine the correct metaphors automatically.

3 Conceptual Metaphor Theory – a linguistic backbone

In order to design and evaluate a system that generates metaphors computationally, we first need to define what a metaphor is. Related work in computational metaphor generation has often limited itself to a superficial definition of a metaphor as the counterpart of "literal" language (cf. [YW19, MLG18]). Yet, this definition is both vague and falls short of explaining metaphors in a structured and comprehensive way.

Aiming to ground our computational work in a structured theory of metaphors, we resort to a definition from cognitive linguistics: Conceptual metaphor theory (CMT). Proposed by George Lakoff and Mark Johnson in 1980, CMT understands a metaphor as a mapping of a source domain onto a target domain[1] [LJ03], [Lak93].

Metaphor scheme	***target domain*** **IS** **source domain**
Metaphor example	*ARGUMENT* IS WAR
Metaphorical expression examples	"She *defended* her claim." "He *attacked* his opponent's argument."

Table 3.1: The metaphor scheme, according to conceptual metaphor theory

3.1 Terminology and structure of metaphors

CMT builds on the idea of domains. A domain is essentially a set of words, phrases, and concepts that belong together semantically. Consider for instance the domain ARGUMENT in table 3.1, which contains words and phrases like "argue", "contradictory statement", as well as concepts like refuting a person's argumentation.

According to CMT, a metaphor is a mapping of one such domain – the so-called *source domain* – onto another domain – the so-called *target domain*. In the above case, the source domain WAR is mapped onto the target domain ARGUMENT. In other words, concepts from ARGUMENT now behave the way concepts from WAR usually do.

One instance of a metaphor, i.e., its occurrence within a sentence, is referred to as a *metaphorical expression*. In table 3.1, the word "defended" evokes the domain WAR, so the sentence "She defended her claim" constitutes a metaphorical expression, an instance of the metaphor which we refer to as ARGUMENT IS WAR. Lakoff and Johnson note that multiple metaphoric mappings can be activated in parallel, thus occurring in the same sentence.

The presence of metaphorical expressions in language by no means implies so-called "poetic language" [LJ03]. Rather, we can find metaphorical expressions in an overwhelming amount of everyday language. They map one

[1] The notion of a source domain that is mapped onto a target domain, thus evoking a metaphor may seem counter-intuitive, since machine translation and similar tasks use the terms in the opposite manner. Nonetheless, we remain consistent with the original terminology from conceptual metaphor theory.

cognitive concept onto another one. While we lack the tools to measure these cognitive mappings directly in the brain, we can find projections of them in language. In that sense, linguistic metaphors are the measurable tip of the iceberg of cognitive mappings. They are one visible layer that allows us to identify some obvious cognitive mappings and reason what the underlying cognitive representations in our brain for these concepts look like and how they interact with each other.

3.2 The invariance principle

While CMT's metaphors are a powerful way of clustering metaphorical expressions, additional rules are at play when producing a correct and meaningful metaphorical expression. Not every concept from a given metaphor's source domain can be mapped onto that metaphor's target domain. Rather, only some of those mappings are suitable. As Lakoff puts it in a hypothesis referred to as the invariance principle:

> Metaphorical mappings preserve the cognitive typology [...] of the source domain, in a way consistent with the inherent structure of the target domain (cf. [Lak93], p. 13).

The invariance principle thus states that *some* cognitive concepts in the source domain behave the way that *some* cognitive concepts in the target domain do, preserving certain semantic relations. These constitute a set of correct mappings that can be applied in metaphorical expressions. Consider for instance the metaphor ARGUMENT IS WAR in figure 3.1. Here, the concept of attacking someone can be mapped onto the concept of criticizing that person's argument. However, the concept of attacking someone cannot be mapped onto the concept of an appropriate argument.

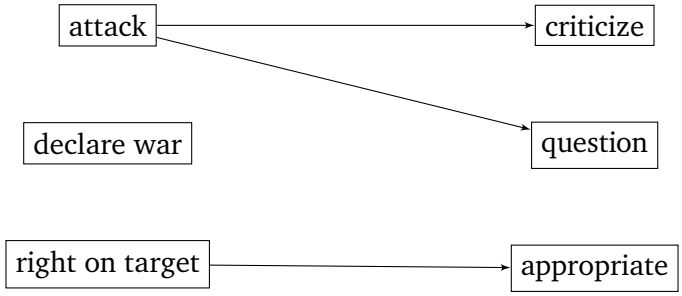

Figure 3.1: Mapping the WAR domain onto the ARGUMENT domain

The invariance principle poses a challenge for computational approaches, since we would expect them to only produce correct and meaningful metaphorical expressions, and not just any mapping from the source to the target domain.

3.3 Criticisms and limitations of conceptual metaphor theory

While CMT is popular in cognitive linguistics, Zoltán Kövecses revisited it in 2008, addressing various lines of criticism [Kö08]. CMT is criticized for lacking a well-defined methodology for identifying metaphors. Instead, Lakoff and Johnson resorted to their intuition for doing so. While valid, this criticism does not impact our work, since we are not looking for an exhaustive corpus of metaphors, but rather for a basis large enough to

trigger a large language model with an additional stimulus. What CMT lacks in methodology, it makes up for in public and transparent data.

Kövecses also points out inner contradictions in CMT's aspirations to account both for universal explanations as well as cultural specificity. He argues that CMT needs to be refined in order to address such criticism. Yet, once again, for computational purposes, CMT contains a sufficient amount of structure to ground our approach in. Furthermore, we expect large language models to grasp context specificity to some extent and to thus implicitly handle such shortcomings of CMT.

Finally, CMT is derived from studying the English language. Gordon et al. report success in applying CMT to Spanish, Farsi, and Russian as well [GHMM15]. Although CMT is grounded in the cognitive system of humans, and therefore assumed to work across languages, which are structured in myriad ways, it is worth noticing that CMT has only been applied successfully to a handful. We go into more detail about this limitation and others in chapter 9.

4 Lexical resources for metaphors

Machine learning is a data-driven discipline. To successfully train a model to generate metaphorical language, we need large amounts of data representing that metaphorical language. Previous approaches have found a lack of metaphorical data to be a tight constraint [CZMP21, SRG20].

Having introduced conceptual metaphor theory (CMT) in chapter 3, we now turn our attention towards MetaNet, a lexical database built on the principles of CMT. After that, we show how data from MetaNet can be complimented with additional data from FrameNet, a larger lexical database. In chapter 6, we leverage the high quality data, structure and annotations from MetaNet and FrameNet to build our own data set of metaphorical sentence mappings.

4.1 MetaNet: An entry point to metaphor data

MetaNet is a lexical database of conceptual metaphors, compiled in the 2010s by the International Computer Science Institute at University of California, Berkeley [DHS15]. Funded mostly by the U.S. Intelligence Community, MetaNet focuses on metaphors' effects in public discourse. At a very broad glance, MetaNet consists of

- more than 550 *frames*, corresponding to domains in conceptual metaphor theory
- more than 650 *metaphors*, i.e., mappings from a source frame to a target frame (cf. §3)

Both the set of frames as well as the set of metaphors are in English. As noted earlier, the MetaNet project also attempted to compile corresponding databases for Spanish, Farsi and Russian [GHMM15], yet the results were not publicly available at the time of writing.

Since MetaNet was compiled by cognitive linguists during years of manual work, we consider it a high quality database and use the metaphors identified in MetaNet as the anchor of the data set we build.

Access MetaNet can be browsed via a web site[1]. While user-friendly, an access paradigm based on browsing does not allow for efficiently automated retrieval of data from MetaNet. Unfortunately, the project provides no means of programmatic access to the database, such as an Application Programming Interface (API) or a querying tool.

We respond to this challenge by writing a web-scraper in Python. We scrape the data for all metaphors and frames and store the result in JavaScript Object Notation Format (JSON). This format allows for efficient

[1] `https://metaphor.icsi.berkeley.edu/pub/en/index.php`, accessed on March 23, 2021

automatic querying on a local machine, circumventing the latency of web queries, but is also readable for humans[2].

Representation and structure Conceptual metaphor theory and therefore also MetaNet almost seamlessly fit into the object-oriented programming paradigm. More precisely, both metaphors and frames can be understood as instances of a class. As shown in figure 4.1, each element in the Metaphor class is defined by a source frame (referred to as *source domain* in CMT) and a target frame (referred to as *target domain* in CMT). We represent these frames as objects as well, belonging to the class MNFrame.

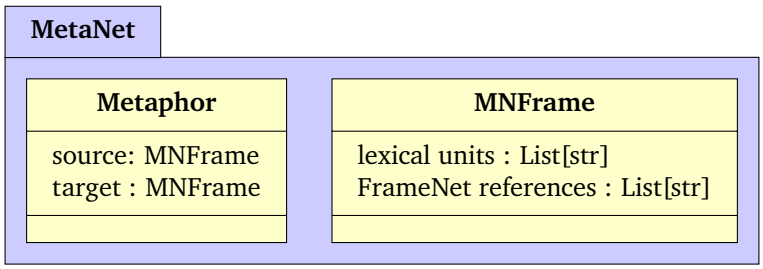

Figure 4.1: Class diagram of all the relevant data that we extract from MetaNet

For each frame in MetaNet, we scrape a corresponding list of lexical units, if available. These lexical units form a vocabulary belonging to that frame. In addition, we also scrape a list of references to frames in FrameNet, a secondary database, if available.

4.2 Linking MetaNet to FrameNet

While MetaNet is a rich resource for metaphorical structure, it contains very little authentic text data from human sources, such as blog posts, dictionary entries, newspaper articles etc. Only occasionally do the authors provide an example sentence for a metaphor.

To tackle the lack of real-world text data, we link MetaNet to FrameNet. FrameNet is a much larger lexical database and was compiled in an over 20 years long effort by the same institution that built MetaNet, the International Computer Science Institute at University of California, Berkeley [BFL98], [Bak15]. While FrameNet does not connect to the concept of metaphors on any level, it also organizes language and vocabulary in frames (or domains, as CMT would put it). FrameNet is structured as a set of relational databases, containing

- more than 1,200 *frames*
- more than 13,000 word senses, i.e., *lexical units*
- more than 200,000 annotated *example sentences*, illustrating the use of individual lexical units

What FrameNet lacks in metaphorical underpinnings, it makes up for with data. As depicted in figure 4.2, we connect MetaNet frames to FrameNet frames where possible, allowing us to expand a MetaNet frame's vocabulary to more lexical units and to retrieve a set of real-world example sentences for that frame.

[2]For instance, the web browser Mozilla Firefox comes with the ability to render and display JSON files in a user-friendly manner.

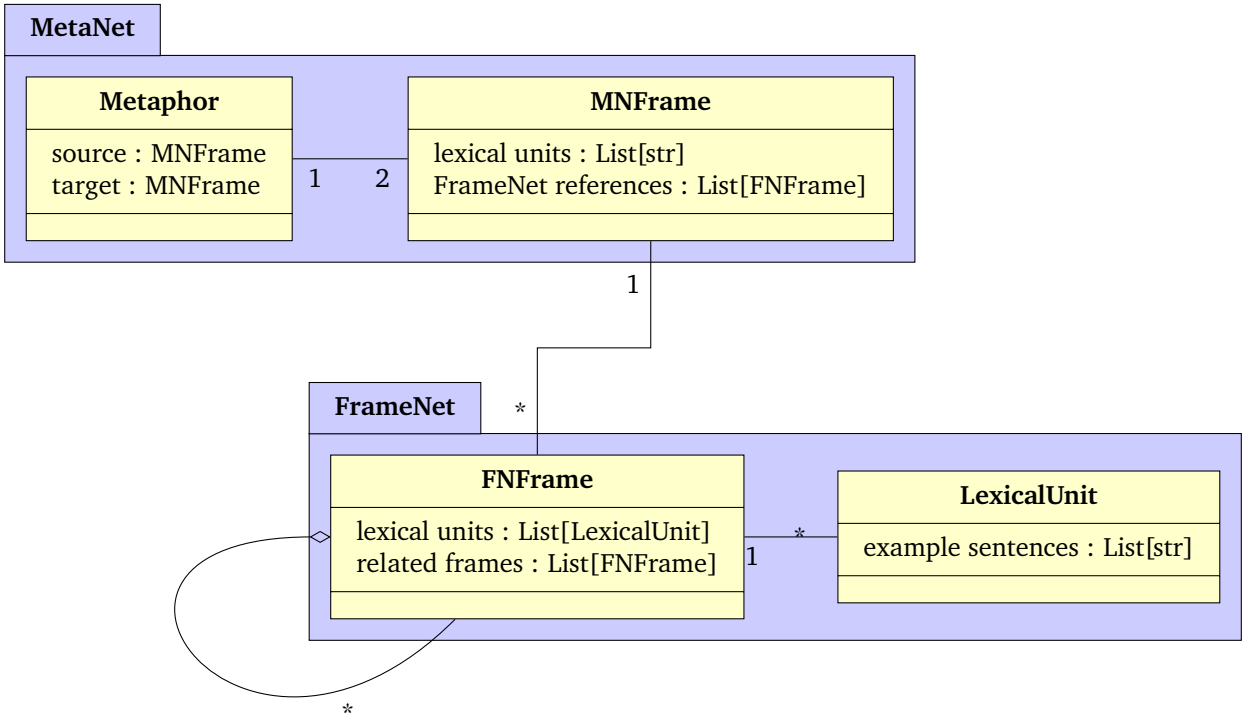

Figure 4.2: Relations between MetaNet and FrameNet data

Access Just like MetaNet, the FrameNet database can be browsed via a respective web site[3]. Additionally, Schneider and Wooters introduced an open source Python API, integrated into the Natural Language Toolkit [SW17], providing programmatic access to FrameNet[4]. We resort to this very efficient and user friendly method of plugging FrameNet into our Python code in chapter 6.

[3] `https://framenet.icsi.berkeley.edu`, accessed on March 23, 2021
[4] `https://github.com/nltk/nltk/blob/3.5/nltk/corpus/reader/framenet.py`

5 Metaphor generation as a computational task

As indicated by the task name "metaphor *generation*", we frame metaphor generation as a computational task from the field of natural language generation (NLG). Broadly speaking, NLG refers to transforming structured data (e.g., mathematical graphs, coordinates , temperature values, but also text) into textual data (e.g., a weather report).

In this thesis, we propose a task for metaphor generation, i.e., activating metaphors in an input text, in accordance with conceptual metaphor theory. The resulting metaphorical text can have various advantages when compared to its "raw" baseline. Lakoff and Johnson point out that metaphorical expressions ease the understanding of abstract ideas in text [LJ03]. In an extensive NLG survey, Gatt and Krahmer suggest that the main function of conceptual metaphor is to enhance the readability of text [GK18], i.e., to make it easier for a reader to comprehend a text. Although not guided by CMT, Yu and Wan see enhancing performance on NLP tasks in general as one goal of metaphor generation [YW19]. Chakrabarty et al. picture creative writing assistance as a potential downstream application [CZMP21]. Hence, metaphors can be seen as making a text more easier to understand, more engaging or more appealing to the reader, opening a wide field of applications for metaphor generation.

Def.: Controlled metaphor generation We define controlled metaphor generation as the function g_c below:

$$\begin{aligned} g_c : \Sigma^* \times D &\rightarrow \Sigma^*, \\ s_{in}, d &\mapsto s_{out} \end{aligned} \tag{5.1}$$

where Σ is an alphabet and D a set of source domains, whose metaphor mappings are to be activated.

Given an input text sequence s_{in} and a source domain d, the task is to produce a corresponding output text sequence s_{out}, in which suitable metaphoric mappings from the given source domain are activated. E.g., given the input text sequence *"Money was transferred to her bank account"* and the source domain *"Liquid"*, a valid output text sequence could be *"Money flowed to her bank account"*, since Money is now conceived as behaving like a liquid.

As [GK18] acknowledge, NLG tasks do not tend to use a uniform input or output format. Yet, by referring to the task definition in equation 5.1, we try to be as consistent as possible with related work such as [CZMP21] and [SLBJ17], while also formulating the task in accordance with conceptual metaphor theory.

Def.: Free metaphor generation We define free metaphor generation as the function g_f below:

$$\begin{aligned} g_f : \Sigma^* &\rightarrow \Sigma^*, \\ s_{in} &\mapsto s_{out} \end{aligned} \tag{5.2}$$

where Σ is an alphabet.

The setup is very similar to equation 5.1, yet for this task, we do not explicitly tell the language model from which source domain we expect metaphors to be activated. Rather, we hypothesize that it *implicitly* learns information about which metaphorical mappings can be activated.

Controlled vs. free metaphor generation Both controlled and free metaphor generation enable interesting experiments. Controlled metaphor generation allows us to tell a model which source domain we wish to activate in an input sentence. We can use the control code d to create novel metaphors, i.e., source-domain mappings that the language model has not seen, and see whether the model is able to make sense of novel metaphors and outputs meaningful text. This might allow us to think of the input sentence in novel ways.

Free metaphor generation on the other hand, allows for ease of use and ambiguity. Users without linguistic knowledge of conceptual metaphor theory will be able to use it nonetheless, since no control code for a source domain is needed. It could be applied for writing assistance, or for enhancing text quality a posteriori. What is more, free metaphor generation leaves it up to the language model to decide which of potentially multiple source domains it activates, leading to different behavior than is expected of controlled metaphor generation.

6 Building a set of sentence mappings

With both a theory of what metaphors are (§3), high quality lexical data bases (§4), and a computational task of metaphor generation (§5) on our hands, we create a novel data set of metaphorical sentence pairs. Each sentence pair in this data set consists of an input sentence and a corresponding output sentence, in which metaphors are activated While our novel data set allows us to fine-tune a language model for metaphor generation, it may also be of interest for other research. Informed by conceptual metaphor theory and based on high-quality expert annotations, as well as consisting of more than 360.000 mappings, our data set is uniquely suited for data hungry applications in natural language processing.

6.1 Overview

We ground our metaphor data set in the metaphors identified in MetaNet (§4.1), a lexical resource built by human experts in cognitive linguistics. For each metaphor we retrieve a list of example sentences which belong to FrameNet frames (§4.2 directly linked to that metaphor's target frame. We then mask one or more words in each example sentence with a mask token. Querying MetaNet and FrameNet, we retrieve a list of replacement options – a vocabulary – from the metaphor's source domain. We extend this list of options by running the FitBERT delemmatizer[1] [HS19] over it . Finally, we use the BERT language model [DCLT19] to pick the most probable replacement option among our vocabulary list and use it to fill the masked slot in the example sentence. This way we create a sentence pair, mapping an input sentence in the target domain to an output sentence, with metaphorical mapping(s) from the source domain activated. Figure 6.1 illustrates this workflow.

Example Consider the Metaphor MONEY IS A LIQUID[2], from MetaNet, also depicted in figure 6.2. We retrieve the hypothetical example sentence "Money was transferred to her bank account" from FrameNet, which contains words from the target frame MONEY. We then mask "was transferred", since we would like to replace it with a metaphorical option from LIQUID. Hence, we have "Money [MASK] to her bank account". Then, we retrieve a list of replacement options from MetaNet and FrameNet, for instance, "flow", "water", and "soft drink". Now, we use the FitBERT delemmatizer to add other word forms, such as "flowed" and "flowing" to the list. Finally, we use BERT to predict the most probable replacement option among the vocabulary list, which is "flowed". Hence, we have the output sentence "Money flowed to her bank account", in which a metaphorical mapping from MONEY IS A LIQUID is activated.

[1] `https://github.com/Qordobacode/fitbert/blob/4b0f2f3f47e5a3fea0ee59d65f71c44b830a93d9/fitbert/delemmatize.py#L23`, accessed April 5, 2021

[2] `https://metaphor.icsi.berkeley.edu/pub/en/index.php/Metaphor:MONEY_IS_A_LIQUID`, accessed April 5, 2021

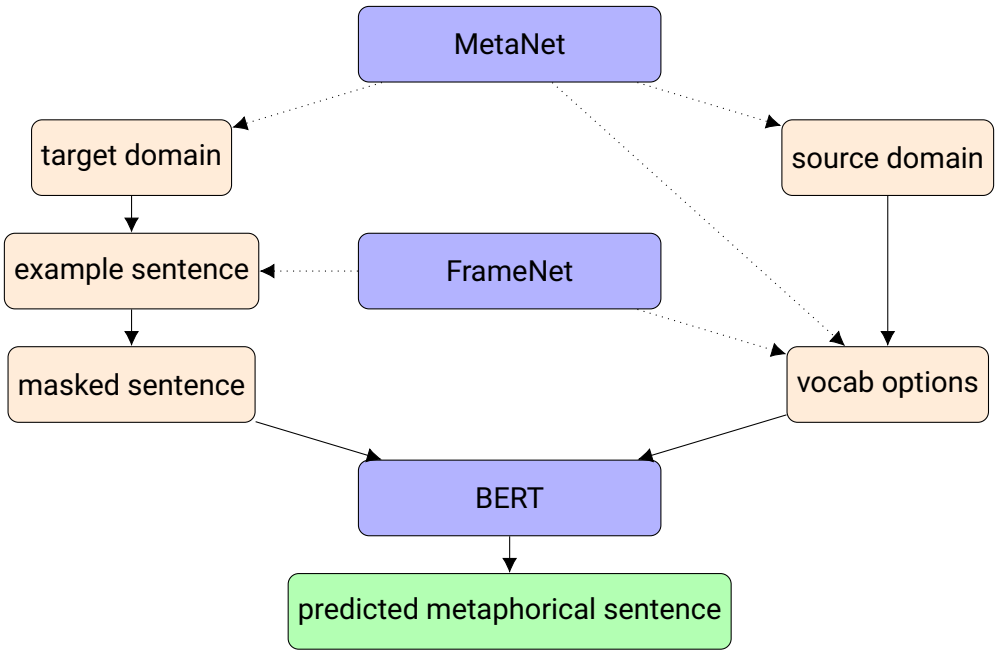

Figure 6.1: The pipeline of creating a sentence mapping

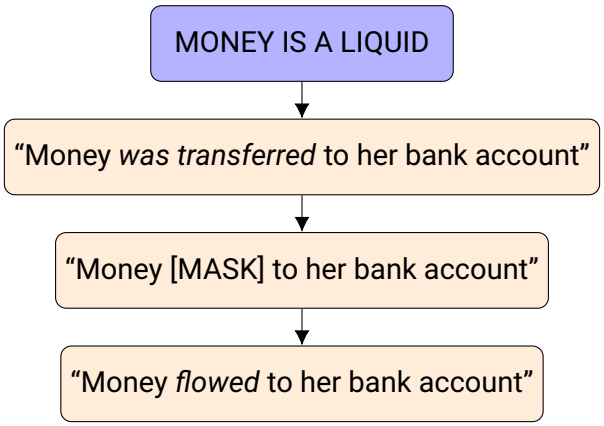

Figure 6.2: An exemplary workflow

6.2 Detailed data flow

In this section, we describe in more detail the individual steps that we took in creating a metaphor data set as well as the considerations regarding those steps.

Retrieving Metaphors from MetaNet We implemented a web scraper in Python, allowing us to retrieve more than 650 metaphors and more than 550 frames from MetaNet (§4.1). Each metaphor consists of a link to a source frame and a link to a target frame. For each MetaNet frame, we scraped a list of lexical units (i.e., the vocabulary belonging to the frame) and a list of links to the corresponding frames in FrameNet. We iterate over all metaphors in MetaNet, generating as many metaphorical mappings as possible.

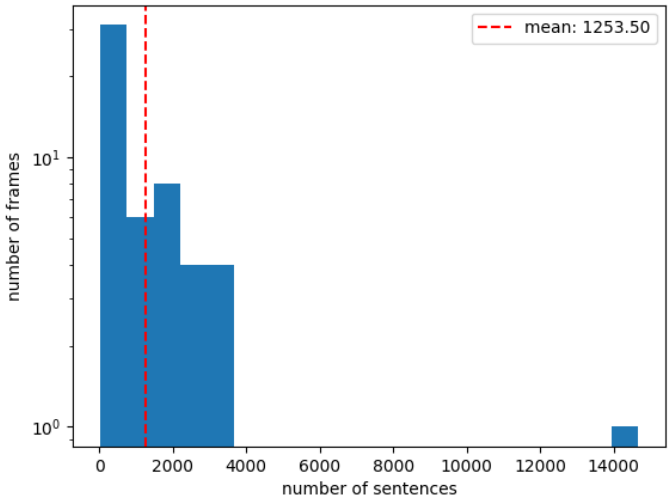

Figure 6.3: Distribution of example sentences per frame (excluding frames with zero examples sentences)

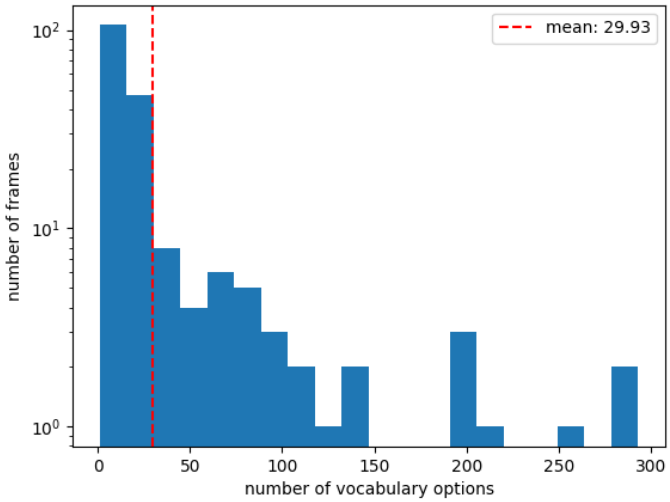

Figure 6.4: Distribution of vocabulary options per frame (excluding frames with zero replacement options)

Retrieving example sentences from FrameNet For each metaphor, we first look at the target frame, from which the example sentence originates. We then leverage the links to FrameNet frames and retrieve all example sentences in FrameNet which belong to the given frames. Additionally, we extend the search space to frames closely related to the previously identified ones. We include frames that are connected via the relations *inheritance*, *subframe*, and *see-also* and also retrieve their example sentences. Since FrameNet contains many more example sentences per frame than MetaNet and since it is much easier to query via a Python API[3], we refer exclusively to FrameNet for example sentences.

As shown in figure 6.3, the average MetaNet target frame can be linked to more than 1000 example sentences. Moreover, we observe that the amount of example sentences per target frame is widely spread, over-representing some metaphors significantly in the resulting data set. This is due in part to a hierarchical organization of frames, with higher-level frames, e.g., SUBSTANCE, including all links from lower-level frames, e.g., LIQUID.

Masking words in the example sentences For each example sentence, we also retrieve an annotation from FrameNet, indicating the position of a target word. We then mask this target word with a mask token. In theory, multiple words can be masked in a sentence, yet almost always we mask just one, since FrameNet usually contains only one respective annotation. Future work might benefit from masking various words in a sentence, potentially leading to more complex metaphorical expressions. Masking various words might also allow various metaphors to be activated in the same sentence.

Building a list of replacement options We use the lexical units from a metaphor's MetaNet source frame and related FrameNet frames to build a list of replacement options. The scope and length of this list is crucial. If chosen too narrow, the output sentences may result strange or unfitting, e.g., due to unsuitable word forms, for instance "Money *water* to her bank account". The broader the scope, the less likely it becomes that the

[3] `https://github.com/nltk/nltk/blob/3.5/nltk/corpus/reader/framenet.py`, accessed April 5, 2021

most probable replacement option is still metaphorical, since we suspect BERT to favor non-metaphorical language. Hence, we limit the replacement options to those from the MetaNet source frame and related FrameNet frames. To tackle the potential problem of incompatible word forms, we incorporate the FitBERT delemmatizer, which expands a word to a list of words that are rooted in the same lemma. E.g., “flow” is delemmatized to “flow”, “flowed”, “flowing”.

As shown in figure 6.4, the average MetaNet source frame can be linked to almost 30 vocabulary options. As with the distribution of example sentences per frame, we observe a high variance in the number of vocabulary options per frame. Again, this is due in part to a hierarchical organization of frames, with higher-level frames, e.g., SUBSTANCE, including all links from lower-level frames, e.g., LIQUID.

Predicting metaphoric mask replacements We turn to the BERT language model for predicting the most probable and thus most suitable replacement word for the masked tokens in the example sentences. Having experimented with FitBERT [HS19] but found it to perform poorly regarding time constraints, we took inspiration from it and implemented FitterBERT, our own predictor, with a more than 1000 fold increase in time efficiency when compared to FitBERT on our machine.

FitterBERT can be understood as a filter layer on top of the BERT base model. While BERT base is used to predict the most probable replacement option for a mask token in a sentence, FitterBERT filters these predictions for just the ones in the list of replacement options and picks the most probable among them.

Putting it all together Having performed all the steps above, we were able to build a data set containing more than 360.000 pairs of input and output sentences, in which metaphorical mappings are activated. Uniquely, the data set is strongly grounded in conceptual metaphor theory.

6.3 Final structure and characteristics of the data set

We release our metaphorical data set for public use. We store it in the TSV file format, since all major machine learning libraries, such as PyTorch or TensorFlow, support it. Furthermore, easy to integrate into most other tools, and easy for humans to read, even with basic text editors. Each of the more than 360.000 entries in the data set contains the following fields:

- ID
- source frame (MetaNet)
- target frame (MetaNet)
- input sentence
- output sentence
- position(s) of replaced word(s) in input sentence
- word type(s) of replaced word(s)

Consisting of text only, our metaphor data set is approximately 20 MB large when compressed as a ZIP file.

7 Fine-tuning a large language model

In this section, we fine-tune T5, a large language model with the set of metaphorical sentence pairs that we compiled in §6. We expect T5 to pick up the patterns from our metaphorical training data and to learn to apply them to the various magnitudes more of language data that is already contained in T5. Ideally, T5 is then able to generalize to text that was not covered in the training pairs, also activating metaphors in this "unseen" text. For a detailed evaluation of our fine-tuning experiments, refer to §8.

7.1 T5: The Text-To-Text Transfer Transformer

T5, a shorthand for "Text-To-Text Transfer Transformer", is a language model published by Google in 2020 [RSR+20] and designed to explore transfer learning possibilities in natural language processing. T5 follows the currently popular and successful transformer architecture [VSP+17] and is trained to receive text prompts like "*translate English to German: I like cats.*" as inputs and to return text like "*Ich mag Katzen.*" as an output or prediction. Given this setup, T5 can easily be fine-tuned to a novel task by introducing that task in the above manner. For instance, we use prompts like "*activate metaphors from Liquid: Money was transferred to her bank account.*" to train T5 on the task of controlled metaphor generation. For free metaphor generation we do not indicate a source domain, i.e., removing "*from Liquid*" from the prompt.

Training T5 was trained on the Colossal Clean Crawled Corpus[1] (C4; [RSR+20]), consisting of more than 750 GB of English text, using self-supervised masked language modeling. The language model thus learns by covering spans in the text from C4 with a mask token and then tries to predict which words were covered by that mask token. Since the model knows the original text, it can then correct or reward its own predictions.

	T5	**BART**	**CTRL**
training data language(s)	English	English	English + DE, ES, FR Wikipedia
training data size	750 GB text	160 GB text	140 GB text
baseline model size	220M parameters	139M parameters	1.6B parameters
architecture	transformer	transformer	transformer
developer	Google	Facebook AI	Salesforce Research
availability	GitHub, Hugging Face	GitHub, Hugging Face	GitHub, Hugging Face

Table 7.1: Candidate models for fine-tuning

Among various candidate models (cf. table 7.1), we picked T5-base, the T5 baseline model, which due to its comparatively small size can be run on a single GPU. T5-base is also a good fit with regard to its state-of-the-art

[1] available at `https://www.tensorflow.org/datasets/catalog/c4` and `https://huggingface.co/datasets/c4`, accessed on May 5, 2021

performance on a set of NLP tasks, the ease with which it can be integrated, as well as the availability of extensive documentation and resources.

Model size T5-base contains roughly 220 million trainable parameters, which collectively determine the state of the language model (cf. §3.6, [RSR+20]). It is comparable in size to the respective BERT baseline model (140 million parameters) [DCLT19] and the respective BART baseline model (139 million parameters) [LLG+20].

Architecture T5-base implements an encoder-decoder architecture, in a configuration that is similar to the BERT baseline model. Encoder and decoder both consist of 12 blocks, with each block containing self-attention, optional encoder-decoder attention, and a feed-forward network. The feed-forward networks consist of a dense layer with an output dimension of 3072, followed by a ReLU activation function and another dense layer. T5-base contains roughly twice as many parameters as BERT-base, since each block T5-base block contains two dense layers rather than one, as in BERT-base (cf. §3.1.1, [RSR+20]).

Availability All related code from the developers of T5 is publicly available on GitHub[2]. The pre-trained model T5-base can be downloaded directly from Google Cloud Platform[3], but also via the Hugging Face API[4], allowing for seamless integration into software development with PyTorch or TensorFlow, two popular deep learning frameworks. We use the Hugging Face `transformers` library [WCD+20] with PyTorch, integrating T5-base by means of the class `transformers.T5ForConditionalGeneration`, which gives us the T5-base model with a language modeling head on top.

Other candidates While we chose to fine-tune T5, two other large language models particularly stood out as fruitful candidates, which may be interesting for further research. BART [LLG+20] is very similar in terms of architecture, use case and linguistic context, yet it is trained on considerably less text data and contains roughly half the amount of parameters in the baseline model. Like T5, BART can be fine-tuned for text generation tasks. CTRL [KMV+19] stands out as a recent large language model, specifically designed for controlled text generation. In contrast to T5 and BART, it is not trained exclusively on English text, but multilingual, since the German, Spanish and French Wikipedia are also contained in CTRL's training data. Unlike T5 and BART, CTRL comes in just one size, featuring 1.6 billion parameters and therefore vastly exceeding the model size that we are working with.

Limitations While T5 has pushed state-of-the-art performance in various natural language processing tasks, it is prone to err in cases that fall outside the scope of its training data. T5 was trained on the Colossal Clean Crawled Corpus – C4 [RSR+20]. C4 is comprised of unlabelled text data at a previously unseen scale, totalling roughly 750 GB. Intended to contain only English text, C4 limits us to fine-tune our language model on *English* metaphor generation.

What is more, Dodge et al. (2021) find that English text data from C4 is distributed highly unequally [DSM+21]. While more than half of all text data stems from US servers, only 3.4% that amount comes from

[2] `https://github.com/google-research/text-to-text-transfer-transformer`, accessed on May 5, 2021
[3] `https://console.cloud.google.com/storage/browser/t5-data/pretrained_models/base`, accessed on May 5, 2021
[4] `https://huggingface.co/t5-base`, accessed on May 5, 2021

Indian servers, even though India is the country with the second largest population of English speakers in the world. In terms of content, 4 of the top 10 websites, when ranked by the number of contributed tokens, belong to news outlets from the US. 1 out of these ten belongs to a UK news outlet and none belong to news outlets from other countries with English speaking populations, such as India, Pakistan, Nigeria, or the Philippines. Moreover, C4's procedure for blocking text of low quality disproportionately removes African American English as well as Hispanic-aligned English (cf. [DSM+21]), hence not giving the language model a chance to appreciate these dialects. We recognize that our work can be expected to propagate the biases associated with the selection of training data forward, at least to some degree.

T5 also imposes rather technical limitations, such as sequence length. Our setup can handle input sequences that are at most 512 tokens long.

7.2 Experiments: Controlled and free metaphor generation

We randomly shuffle the metaphor data set from §6, which contains more than 360,000 sentence pairs, and then split it into a training set (90%), a development set (5%) and a test set (5%). We use a fixed seed, allowing others to reproduce the training process. We then implement a customized PyTorch `Dataset` class, namely `MetaphorDataset`, containing the input text sequences in the format that T5 requires. Namely, controlled metaphor generation inputs follow the template "*activate metaphors from <SOURCE DOMAIN>: <INPUT SENTENCE>*", while free metaphor generation inputs follow the template "*activate metaphors: <INPUT SENTENCE>*", without the control code for a source domain.

We download the T5-base model with a language modeling head as well as a corresponding tokenizer via the Hugging Face `transformers` Python API [WCD+20]. For both controlled and free metaphor generation, we use the high level training interfaces provided by Hugging Face's `Trainer` class. Table 7.2 provides an overview of the hyper parameters that we have chosen. We leave all other hyper parameters at the default choices specified in `Trainer.TrainingArguments`[5].

no. epochs	3
batch size	24 (controlled), 12 (free)
weight decay	0.0001

Table 7.2: Hyper parameters for controlled and free metaphor generation

We execute both experiments on an NVIDIA v100 32 GB device. Following calls for efficiency reporting [BGMMS21], we use `codecarbon` [SGJ+21] to track emissions during fine-tuning, which can be found in table 7.3.

	controlled generation	**free generation**
energy consumption	3.55 kWh	3.34 kWh
equivalent CO_2 emissions	2.47 kg	2.26 kg

Table 7.3: Emissions during fine-tuning

[5] For more details, refer to the online documentation: `https://huggingface.co/transformers/main_classes/trainer.html#trainingarguments`, accessed on May 6, 2021

Although fine-tuning methods like gradual unfreezing or adapter layers exist, we fine-tune all parameters in T5-base. Given that metaphors can be seen as fundamental to language rather than a high-level task, we consider this appropriate.

7.3 Results

We produce two independently fine-tuned versions of T5-base, one version for controlled metaphor generation and one for free metaphor generation. We plan on releasing both models to the public in an easy-to-use manner, via Hugging Face's Model Hub [WCD+20].

8 Evaluation

In this section we evaluate our experiments from §7, where we fine-tuned T5 in two setups for controlled and free metaphor generation. Precisely, *we evaluate, whether the models have learned to generate conceptual metaphors*. We refer to three criteria to determine whether a model generates conceptual metaphors:

- metaphoricity: The output sentence is intuitively perceived as metaphorical.
- fluency: The output sentence is perceived as fluent, i.e., linguistically sound.
- semantic similarity: The metaphorical paraphrases from input to output sentence are similar in meaning, i.e., they behave in accordance to Lakoff and Johnson's Invariance Principle [LJ03].

We evaluate our models on various criteria rather than one, because no single default criterion has been established yet to judge metaphor generation [GK18, CZMP21, SRG20]. Various facets are at play when a good metaphor is generated and we deem metaphoricity, fluency, and semantic similarity particularly important. Related work has been evaluated on similar criteria [CZMP21, SRG20, SCP$^+$21], allowing for comparisons.

Consider the following sentence pair, which constitutes a correct conceptual metaphor according to the above criteria.

"Activate metaphor from Liquid: Money was transferred to her bank account".
$\mapsto$ "Money flowed to her bank account."

The output sentence is metaphorical, it is fluent, i.e., easy to read and understand, and it correctly paraphrases the previous words "was transferred", i.e., reflecting a suitable concept from the domain "Liquid". We therefore consider the given sentence pair a successful instance of metaphor generation.

8.1 Measuring successful metaphor generation

We use a set of roughly 15,000 previously unseen sentence pairs to evaluate our models – 5% of the data set that we created in §6. While an evaluation set of this size allows for statistically meaningful analysis, it makes manual evaluation unfeasible, calling for automated evaluation metrics.

Unfortunately, we are not aware of an automated metric that is able to score the first two criteria, i.e., metaphoricity and fluency, for our purposes. We therefore turn to manual evaluation on a subset of 100 sentence pairs. For the third criterion, semantic similarity, we also implement SBERT [RG20], an automated metric for semantic sentence similarity, as a proxy. If an input sentence is paraphrased correctly to form a metaphorical output sentence, we would expect semantic sentence similarity to be high, but not perfect.

We point out that metaphor generation has been struggling to find automated metrics that correlate well with human judgement [GK18]. While [CZMP21] compute the BLEU score, a metric for machine translation, we choose not to do so, given that it is not applicable to metaphor generation [GK18]. For similar reasons, we refrain from computing perplexity as a proxy for quality, given that it favors language that is more prevalent in its reference corpus, which we do not assume metaphorical language to be.

8.2 Methodology

We report two metrics based on SBERT [RG20] on our full evaluation data set, serving as proxies for semantic similarity. We conduct an annotation study on a subset of 100 evaluation samples that reflect the linguistic ambition of our approach particularly well, 50 of which stem from "specific" metaphors, with the other 50 stemming from "general" metaphors, as per the MetaNet labels. We also evaluate on an unseen data set that [SCP$^+$21] compiled in a different approach. Their data set consists of 150 sentence pairs that our models have not seen yet, i.e., with unseen domain labels or unseen domain mappings as well as unseen sentences. While we use SBERT to score semantic similarity automatically, we conduct an annotation study to score the metaphoricity, fluency and semantic similarity by human judgement on a total of 250 sentence pairs, referred to as $eval_{human}$.

We report results from automated and human evaluation across the full evaluation sets and compare our free (CM-T5-free) and controlled (CM-T5-ctrl) metaphor generation models to MERMAID [CZMP21], CM-BART [SCP$^+$21], and MetMask [SRG20]. We then report results by metaphor level, and compare performance on seen and unseen metaphors. Informed by [GK18], we control for correlations between human judgement and automated scores, assessing the usefulness of our automated metrics.

8.3 Results from automated evaluation

In addition to our development set – which we used for small adjustments in the training process – we set aside an unseen evaluation set, consisting of 5% of the sentence pairs that we generated in §6. This evaluation set, $eval_{full}$, consists of 15833 sentence pairs, allowing us to evaluate our models in a statistically meaningful manner.

We compare our models for controlled (CM-T5-ctrl) and free metaphor generation (CM-T5-free) against MetMask [SRG20], MERMAID [CZMP21] and CM-BART [SCP$^+$21], computing the dis and rel metrics, which are based on Sentence-BERT embeddings [RG20]. We use both metrics as a proxy for semantic similarity, cautioning that no single metric is expected to capture metaphor generation well and that automated metrics do not always correlate well with human evaluation [GK18].

Definitions of our metrics Let $(gold_{in}, gold_{out})$ be our gold sentence pair from the evaluation data, which we use as a reference. Let $pred$ be the model prediction for the output and Σ be the alphabet over which all three sentences are defined. Let cos denote the cosine similarity between two vectors, i.e., embeddings. The evaluation metrics are then defined as follows.

$$dis : \Sigma^* \times \Sigma^* \to [0, 1],$$
$$gold_{out}, pred \mapsto 1 - cos(SBERT(gold_{out}), SBERT(pred))$$

$$rel : \Sigma^* \times \Sigma^* \times \Sigma^* \to [0, 1],$$
$$gold_{in}, gold_{out}, pred \mapsto |cos(SBERT(gold_{in}), SBERT(gold_{out})) - cos(SBERT(gold_{in}), SBERT(pred))|$$

We first use Sentence-BERT (SBERT)[1] to compute semantic sentence embeddings for the sentences. Then we compute the cosine similarity between those embeddings. By taking the complement to that cosine similarity, dis gives us the cosine distance between $gold_{out}$ and $pred$. The closer to zero that value is, the more similar the meaning of both sentences.

rel gives us the relational distance. The closer to zero, the better the semantic similarity of the gold sentence pair matches that of the predicted sentence pair.

Overall results We report the mean dis and rel scores across $eval_{full}$ in table 8.1. We find that our controlled approach significantly outperforms the free approach as well as all other models in this comparison on both metrics. Controlled predictions thus more closely capture the intended meaning than free predictions. This finding indicates that control codes influence text generation significantly, leading our controlled model to paraphrase in different ways than the free generation model would.

Model	dis	rel
MetMask	.191	.094
MERMAID	.147	.087
CM-BART	.085	.047
CM-T5-ctrl	**.041**	**.024**
CM-T5-free	.081	.057

Table 8.1: SBERT-based dis and rel scores, compared to state-of-the-art models, $n = 15833$ samples

However, the reported metrics need to be taken with a grain of salt, given that in some cases the models do not paraphrase at all. That is, they merely output the input sentence with only slight changes, such as punctuation. The dis and rel scores are thus better suited to compare our two models than to provide an absolute quality score.

Results by metaphor level MetaNet metaphors, on which our models were trained, include a "level" attribute. A metaphor can either be labelled as "specific", "general" or not be labelled at all.

In figure 8.1, we report the distributions of dis and rel values by metaphor level, i.e., separately for "specific" and "general" metaphors. We also report the overall distribution, i.e., across $eval_{full}$ as a whole for comparison.

[1] Precisely, we use the pre-trained SBERT model `stsb-roberta-large`, cf. `https://huggingface.co/sentence-transformers/stsb-roberta-large`, accessed on May 18, 2021

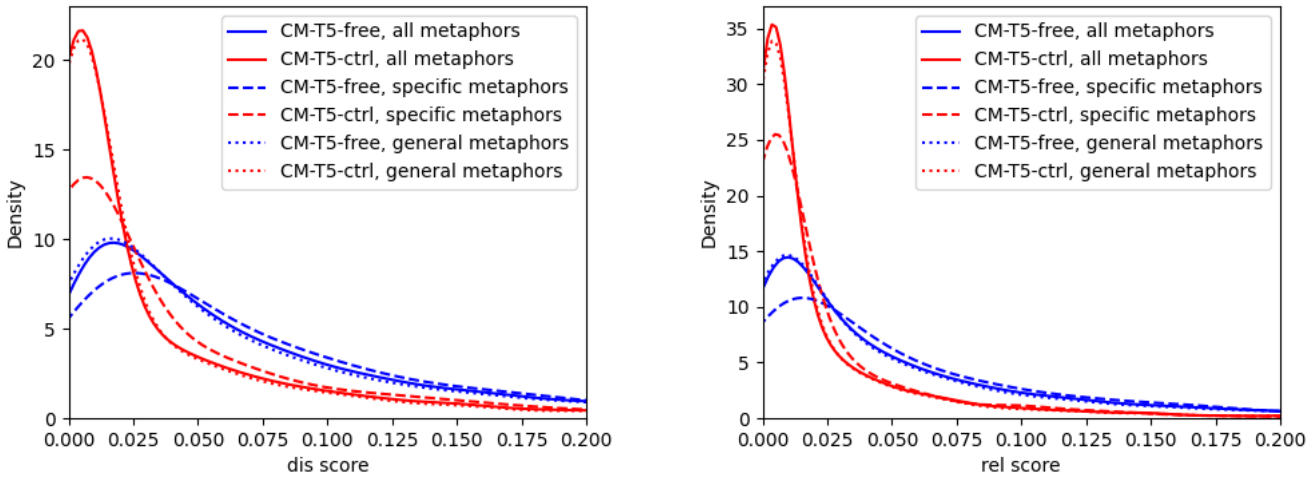

Figure 8.1: dis (left) and rel (right) scores' distribution, by metaphor level. Red lines relate to our controlled model; blue lines concern our free model.

We find that controlled predictions exhibit less variance in semantic similarity, across all categories, making controlled metaphoric paraphrases more semantically similar than free paraphrases, on average. Given that controlled generation is intended to specifiy more closely what kind of metaphor we want, this finding confirms the benefits of this approach. The finding also shows that our free generation model learns different paraphrases than the controlled model, even when trained on the same data. The free model's higher variance in dis and rel scores shows that it is more prone to generate rather far-fetched metaphorical paraphrases, compared to the controlled generation model. We can thus confirm that both controlled and free generation have their individual benefits and use cases.

We also find that specific metaphors, both controlled and free, exhibit considerably more variance in semantic similarity than do general metaphors. On average, they are less similar. General metaphors' behavior on the other hand does not deviate significantly from the overall behavior in the evaluation set. Assuming dis and rel to be meaningful proxies for paraphrase quality, this indicates less reliability in generating specific metaphors regarding paraphrase quality. General metaphors' paraphrase quality, on the other hand, does not deviate at all from the overall performance.

8.4 Human evaluation

Given the lack of reliable evaluation metrics, we conducted an annotation study on 250 sentence pairs, a much smaller evaluation set than $eval_{full}$ from section 8.3. Table 8.2 gives an overview of the $eval_{human}$ set, which is composed of general and specific metaphors from our own data, as well as sentence pairs from other corpora, compiled by [SCP+21], §4.1.

Since [SCP+21] use FrameNet frame names as control codes, in contrast to our data, which contains MetaNet frame names, we changed the control codes from [SCP+21] to MetaNet frame names, where possible. Nevertheless, the metaphors, i.e., the mappings from one frame to another, were all novel to our models.

number of pairs	origin	metaphor type	metaphors unseen
50	$eval_{full}$	specific	no
50	$eval_{full}$	general	no
50	Mohammad 2016 Corpus [MST16]	–	yes
50	Gutenberg Poetry Corpus [Jac18]	–	yes
50	Brown Corpus [FK]	–	yes

Table 8.2: Composition of our $eval_{human}$ set

We evaluate the 250 sentence pairs regarding three criteria:

- *semantic similarity* of the sentence pairs
- *metaphoricity* of the output sentences
- *fluency* of the output sentences

We randomly pick 100 sentence pairs from $eval_{full}$, throwing out gold pairs that do not reflect the metaphoric intention of the data set well. In that sense, we filter them to only include samples of interest, but taking into account only the gold pairs, not the predictions. The remaining 150 gold pairs are sampled randomly by [SCP$^+$21].

Annotation study details We hired various crowd workers as annotators, obtaining 5 scores from different persons per criterion and sentence pair. We vetted the annotators, testing their understanding of the criteria on ten example pairs, and proceeded only with those who exhibited a good understanding of all three criteria. We then asked the annotators to score *semantic similarity*, *metaphoricity*, and *fluency* for all 250 evaluation pairs. The scores follow a Likert scale from 1 to 4, where 1 corresponds to a criterion not holding at all and 4 corresponds to a criterion holding fully.

We make the annotation study results publicly available, allowing other research in metaphor generation to be evaluated on them as well.

Performance on native evaluation data First, we evaluate our models for controlled (CM-T5-ctrl) and free metaphor generation (CM-T5-free) on their *native* evaluation data only, i.e., on the 100 pairs from $eval_{full}$. We compare them to the results reported by [SCP$^+$21] for MetMask [SRG20], MERMAID [CZMP21], and CM-BART [SCP$^+$21] from evaluations on those models' *native* data (cf. table 8.3).

While our models do not outperform the others in this setting – CM-BART comes in first regarding metaphoricity – we are able to observe interesting and significant differences in performance between controlled and free metaphor generation. While controlled generation leads to higher metaphoricity than free generation, free generation produces text with better semantic similarity and fluency. This finding confirms our hypothesis that controlled and free metaphor generation respond to different use cases, each having their own benefits. Controlled generation allows us to trigger a particular metaphor at the expense of semantic similarity and fluency, while free generation leaves the decision of which metaphor to generate to the language model, increasing semantic similarity and fluency at the expense of metaphoricity.

Model	metaphoricity	semantic similarity	fluency
MetMask	2.27	–	–
MERMAID	2.56	–	–
CM-BART	**2.72**	–	–
CM-T5-ctrl	2.276	3.315	3.356
CM-T5-free	2.108	**3.575**	**3.561**

Table 8.3: Mean performance on *native evaluation data* only. Scores from 1 (poor) to 4 (great)

Performance on out-of-scope evaluation data We are particularly interested in how our language models perform on unseen data, since we expect most of the language data that will be fed to them not to come from the MetaNet-based data set that we created in §6. Therefore, we evaluate both models – free and controlled – on 150 sentence pairs of out-of-scope data [MST16, Jac18, FK]. Neither have our models seen the input sentences that they are fed, nor have they seen the metaphors that are requested from them. In an overwhelming majority of the cases they have not even seen the individual frame/domain names, since they come from unfamiliar data sets.

Interestingly, our controlled model takes the lead in both metaphoricity and semantic similarity (cf. table 8.4). Given that both models are unfamiliar with the data, controlled generation might prove more reliable in this setting, since we provide it with additional guidance, i.e., a control code for the metaphor's source domain. Nonetheless, free generation shows slightly better results on fluency than controlled generation does, encouraging us to consider both models' advantages.

What is more, both models beat the previous state-of-the-art model CM-BART [SCP$^+$21] in out-of-scope performance regarding metaphoricity, suggesting better generalization capabilities.

Model	metaphoricity	semantic similarity	fluency
CM-BART	2.41	–	–
CM-T5-ctrl	**2.601**	**3.711**	3.195
CM-T5-free	2.446	3.683	**3.378**

Table 8.4: Mean performance on *out-of-scope evaluation data* only. Scores from 1 (poor) to 4 (great)

Performance on mixed evaluation data Having evaluated our models on native and out-of-scope data, we now report the results from evaluation them on both in table 8.1. We continue to observe a trend where controlled generation performs better on metaphoricity, while free generation yields better performance on semantic similarity and fluency.

Model	metaphoricity	semantic similarity	fluency
CM-T5-ctrl	**2.431**	3.458	3.320
CM-T5-free	2.346	**3.645**	**3.395**

Table 8.5: Mean performance on *all, i.e., mixed evaluation data*. Scores from 1 (poor) to 4 (great)

Performance by metaphor level In the spirit of a similar comparison in section 8.3, we analyze how metaphor level ("specific" vs. "general") affects our models' behavior (cf. table 8.6). We find mixed results. General

metaphors clearly score better on metaphoricity. However, in semantic similarity and fluency, our controlled approach performs better on general metaphors while our free approach performs better on specific metaphors.

Although it is unclear how exactly these differences come to be, we are able to show that the metaphor type affects our models' performance and that neither of both models is a clear favorite across the full spectrum of metaphors.

Model	**metaphoricity**		**semantic similarity**		**fluency**	
	specific	general	specific	general	specific	general
CM-T5-ctrl	2.185	2.367	3.238	3.392	3.289	3.424
CM-T5-free	1.971	2.244	3.612	3.538	3.571	3.551

Table 8.6: Mean performance on *specific* vs. *general* metaphors. Scores from 1 (poor) to 4 (great)

8.5 Correlation between automated and human scores

Automated evaluation is a pressing and open problem for the field of natural language generation (NLG) [GK18]. For problems such as metaphor generation, it is hard to find a metric that correlates well with human judgement, which is why this work's evaluation relies largely on human-made annotations rather than the much cheaper automated metrics. In this section we take a look at how well human judgement and our automated metrics – based on SBERT [RG20] – correlate.

Considering the 100 annotated sentence pairs from $eval_{full}$ (cf. table 8.1), we correlate three human metrics – metaphoricity, semantic similarity, and fluency – as well as the mean value of all three with our two automated metrics: dis and rel. We report the resulting matrix of Pearson correlation coefficients in table 8.7.

We find that there is no strong correlation between human and automated metrics with respect to the given sentence pairs, neither regarding individual human metrics nor regarding the mean of all three human metrics. We thus confirm the lack of meaningful automated metrics [GK18] and call for future work on this topic, allowing for speedier testing of metaphor generation systems. We also release our annotated evaluation data, thus allowing further research into metaphor generation to be evaluated without the cost of conducting an annotation study.

We also find that the SBERT-based cosine distance dis correlates best with human-reported semantic similarity, suggesting that among the human metrics at hand, SBERT caters best to the purpose it was designed for: Evaluating semantic similarity of sentences.

Finally, we report a correlation coefficient of 0.050 between the mean of our human metrics and rel, leading us to question rel's usefulness for evaluating metaphor generation.

	dis	rel
metaphoricity	0.168	0.176
semantic similarity	0.220	−0.022
fluency	−0.097	−0.124
mean	0.177	0.050

Table 8.7: Pearson correlation coefficients for human (bold) vs. automated (italic) scores

9 Limitations

In this thesis, we built a novel gold data set for metaphor generation and fine-tuned a large language model to generate metaphors on the basis of that data set. We now address limitations of our work, outlining what cannot be expected from our current metaphor generation models.

Language All of our work is based exclusively on English. While English language resources, such as T5, BERT, MetaNet, and FrameNet, are most widely available and most easily accessible, their exclusive use constitutes a significant limitation. Given that conceptual metaphor theory, the linguistic backbone of this work, works on a cognitive level, independently of one specific language, we are optimistic that it can be transferred easily to other languages. The idea of concepts from one domain behaving in terms of concepts from another domain is by no means limited to the English language. With optimism, we encourage research into metaphor generation for other languages and multilingual settings.

Gold corpus quality Although our methodology for building a gold corpus of metaphorically paraphrased sentence pairs is grounded directly in MetaNet, a high-quality database for conceptual metaphor theory, we find that the gold sentence pairs that this approach produces do not reach human quality. Errors often happen during the word replacement process, in which an input word is replaced with a more metaphorical one. Common errors include meaningless words, such as “the” or “a”, words of the wrong word type or words with unintended meanings, e.g., “date” referring to a fruit rather than a point in time, filling the input word slot. A degree of noise, e.g., unsuitable input sentences or unsuitable word replacement options, can also be found in the gold data. Despite all that, we expect a large-enough language model like T5 to disregard that noise and pick up on successful metaphoric paraphrases primarily, given that they account for valid, “real” language, which a large language model should be able to tell.

Biases Our work is strongly corpus-based. We build our gold corpus on the MetaNet and FrameNet corpora. We also leverage the BERT language model for word replacement, which itself was trained on English text corpora. Finally, we fine-tune the T5 language model, which was trained on even more English text. Large language models especially have recently been shown to contain significant biases which they picked up from the text they were trained on [BGMMS21]. In some cases [BGMMS21] show how biases in the original text are even intensified by language models and lead to harmful model behavior. A recent study of C4 [DSM$^+$21], the text corpus that T5 was trained on, confirms many of the caveats that [BGMMS21] articulate. Consider the following example from our replacement procedure: For the metaphor “Money is a substance”, BERT replaced the word “Money” was in most cases with “paper”. Yet in a sentence that included the word “Muslims” it replaced “Money” with “oil” instead. Behavior like this could well be due to a US-centered world view that is manifested in BERT’s English training data, in which lots of text involving Muslims connects to oil interests, related foreign policy and conflicts in Muslim countries. We caution against expecting unrealistic traits, such as objective, de-biased, or universally good performance, from our fine-tuned models.

Moreover, MetaNet and FrameNet exhibit biases already, manifested in the selection of example sentences, frames, etc., which the creators of these databases made. Funded partly by the U.S. defense community, MetaNet studies the effects of metaphor in public discourse, and vastly overrepresents sentences related to foreign policy, conflict and war.

Reliable evaluation As noted in chapter 8, reliable metrics for metaphor generation have yet to be proposed. Due to the lack thereof, we rely largely on our own intuition as well as a few vetted annotators' judgement for developing our models and judging their performance. While this gives us a good measure of confidence, it falls short of being a reliable metric, or a solid benchmark. We emphasize the limitations posed by the lack of meaningful metrics and call for future research in the field to address them.

10 Future work

Metaphor generation is a relatively small field of natural language processing, compared to other subfields like machine translation or text summarization. Many avenues of research thus remain widely open. We now outline a couple of particularly relevant lines of research regarding our approach to metaphor generation.

Improving gold data While some progress has been made in creating large gold data sets for metaphor generation [SCP+21, CZMP21], there is plenty of headroom, especially regarding the quality of that gold data. Our approach from §6, for instance, could be improved by redesigning the method with which candidate replacement words are gathered for a masked input word. Meaningless words like articles should be eliminated from candidate lists. More attention should be paid to preserving meaning and not including too far-fetched words. It might also be helpful to only consider replacement options of the same word type.

Towards broader paraphrasing Current approaches, including ours, generate metaphors mostly by replacing individual words in an input sentence with other words, evoking a different domain. Future work should explore ways of going beyond the level of word replacement and towards replacing phrases, i.e. multiple semantically connected words. This might require different mechanisms to be used for compiling training data, since phrases are more complex to be identified and replaced than individual words. We could imagine that the attention mechanism [VSP+17] in transformer language models as a starting point for identifying various words that belong together semantically and could all be replaced with more metaphorical ones.

Improving evaluation Evaluation methods are currently both expensive and not very reliable. Future work should explore automated and cheaper evaluation methods that correlate strongly with human judgement. Such methods cannot rely solely on measuring how likely a given text sequence is to occur when compared to others, since this dooms them to favor non-metaphorical language over metaphorical language. Rather, meaningful automated metrics should take into account how words from different linguistic domains interact, possible tagging them on the basis of an ontology like MetaNet or FrameNet.

Implement non-English models Most work in metaphor generation, including ours, focuses on generating English language metaphors. We encourage future work to apply our methodology to other languages as well, hoping to thereby confirm the universal ambition of conceptual metaphor theory. Promising starting points already exist: [DHS15] developed MetaNet for Farsi, Russian, and Spanish, in addition to English. FrameNet also inspired similar works for other languages, e.g. SALSA for German [BEF+06]. These lexical resources can be connected following our methodology, possibly taking the English MetaNet as a starting point and jumping only from English FrameNet frames to corresponding frames in the target language, when data is scarce.

What is more, large transformer language models are also responding to calls for languages other than English to be supported. While we work with T5, which was trained exclusively on English text, [XCR+21] created a multilingual version of T5, covering 101 languages.

11 Conclusion

In this thesis, we conducted transfer learning experiments for the task of metaphor generation, both in a controlled and a free manner. Leveraging MetaNet and FrameNet, two lexical databases, we created a large novel data set of metaphorical sentence pairs, which fellow researchers are welcome to use. We fine-tuned the T5 language model on free and controlled metaphor generation in two separate experiments. We compared both resulting models to related approaches, reporting similar performance overall, but showing that our models generalize better to unseen metaphors, according to our annotation study. Controlled generation produces more metaphorical text, while free generation produces more fluent and semantically similar text, suggesting differing use cases for the individual models. We also analyzed differing performance by metaphor level, showing that specific metaphors' semantic similarity varies considerably more than general metaphors' does. We also compute weak correlation coefficients between our evaluation criteria, calling for future work on automated metrics that correlate better with human judgement.

Overall, we report promising results and hope that our methodology inspires future research, improving performance and yielding additional insights.

12 Bibliography

[ASN06] Keiga Abe, Kayo Sakamoto, and Masanori Nakagawa. A computational model of the metaphor generation process. *Proceedings of the 28th Annual Meeting of the Cognitive Science Society*, 2006.

[Bak15] Collin Baker. Framenet: A knowledge base for natural language processing. 2015.

[BEF+06] A. Burchardt, Katrin Erk, A. Frank, A. Kowalski, and Sebastian Padó. The salsa corpus: a german corpus resource for lexical semantics. In *LREC*, 2006.

[BFL98] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. 1998.

[BGMMS21] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? . *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[CZMP21] Tuhin Chakrabarty, Xurui Zhang, Smaranda Muresan, and Nanyun Peng. Mermaid: Metaphor generation with symbolism and discriminative decoding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4250–4261, 2021.

[DCLT19] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[DHS15] Ellen Dodge, Jisup Hong, and Elise Stickles. Metanet: Deep semantic automatic metaphor analysis. 2015.

[DSM+21] Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, and Matt Gardner. Documenting the english colossal clean crawled corpus. *ArXiv*, abs/2104.08758, 2021.

[FK] W Nelson Francis and Henry Kucera. Brown corpus manual. *Letters to the Editor*, 5(2):7.

[GC19] Katy Ilonka Gero and Lydia B. Chilton. Metaphoria: An algorithmic companion for metaphor creation. 2019.

[GHMM15] Jonathan Gordon, Jerry Hobbs, Jonathan May, and Fabrizio Morbini. High-precision abductive mapping of multilingual metaphors. 2015.

[GK18] Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170, 2018.

[HCC+07] R. Hervás, R. P. Costa, Hugo Costa, P. Gervás, and F. Pereira. Enrichment of automatically generated texts using metaphor. In *MICAI*, 2007.

[HS19] Sam Havens and Aneta Stal. Use bert to fill in the blanks, 2019.

[Jac18] A. Jacobs. The gutenberg english poetry corpus: Exemplary quantitative narrative analyses. *Frontiers Digit. Humanit.*, 5:5, 2018.

[Jon92] Mark Alan Jones. Generating a specific class of metaphors. pages 321–323, 1992.

[KMV+19] Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. Ctrl: A conditional transformer language model for controllable generation, 2019.

[Kö08] Zoltán Kövecses. Conceptual metaphor theory: Some criticisms and alternative proposals. *Annual Review of Cognitive Linguistics*, 6, 2008.

[Lak93] George Lakoff. Contemporary theory of metaphor, 1993.

[LJ03] George Lakoff and Mark Johnson. *Metaphors We Live By*. University of Chicago Press, 2003.

[LLG+20] M. Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, A. Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv*, abs/1910.13461, 2020.

[Mil95] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38, 1995.

[MLG18] Rui Mao, Chenghua Lin, and Frank Guerin. Word embedding and wordnet based metaphor identification and interpretation. volume 1, 2018.

[MST16] Saif M. Mohammad, Ekaterina Shutova, and Peter D. Turney. Metaphor as a medium for emotion: An empirical study. In **SEM@ACL*, 2016.

[OZWI14] Ekaterina Ovchinnikova, V Zaytsev, Suzanne Wertheim, and Ross Israel. Generating conceptual metaphors from proposition stores. *ArXiv*, abs/1409.7619, 2014.

[PSM14] Jeffrey Pennington, R. Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[RG20] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. 2020.

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21, 2020.

[SCH17] R. Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, 2017.

[SCP+21] Kevin Stowe, Tuhin Chakrabarty, Violent Peng, Smaranda Muresan, and Iryna Gurevych. Generating metaphors with conceptual mappings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (accepted)*, online, 2021. Association for Computational Linguistics.

[SGJ+21] Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni. CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing. 2021.

[SLBJ17] Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[SRG20] Kevin Stowe, Leonardo Ribeiro, and Iryna Gurevych. Metaphoric paraphrase generation. *ArXiv*, abs/2002.12854, 2020.

[Ste94] Eric Steinhart. Netmet: A program for generating and interpreting metaphors. *Computers and the Humanities*, 28:383–392, 11 1994.

[SW17] Nathan Schneider and Chuck Wooters. The nltk framenet api: Designing for discoverability with a rich linguistic resource. 2017.

[TN10] Asuka Terai and Masanori Nakagawa. A computational system of metaphor generation with evaluation mechanism. volume 6353 LNCS, 2010.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[WCD+20] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

[XCR+21] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, A. Barua, and Colin Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *ArXiv*, abs/2010.11934, 2021.

[YW19] Zhiwei Yu and Xiaojun Wan. How to avoid sentences spelling boring? towards a neural approach to unsupervised metaphor generation. volume 1, 2019.