FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Amrita Harikrishnan Nair

Unsupervised Open Information Extraction with Large Language Models

Institute of Formal and Applied Linguistics

Supervisors of the master thesis:	doc. RNDr. Pecina Pavel, Ph.D. Prof. Dr. Günter Neumann
Study programme:	Computer Science
Study branch:	Language Technologies and Computational Linguistics

Prague 2024

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations listed under Act No. 121/2000 Sb., the Copyright Act, as amended, in particular, the fact that Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

Firstly, I would like to express my thanks for receiving the Erasmus-LCT scholarship which facilitated my studies at Charles University and Saarland University. I am very grateful for this incredible opportunity.

I would also like to thank my supervisors for their timely input and encouragement. Prof. Pecina was kind, helpful, and motivated me throughout the writing of this thesis. Prof. Neumann first introduced me to this area and always pushed me to do my best. Thank you so much!

This thesis would not have been possible without the support of my family and friends. Thank you, Rupal and Daniela; I would not have survived the last few days without you.

Thank you to my family (Mama, Baby, Ravi and Rahul Bhaiya) for all your support over the last few years.

Last but not least, thank you, Amma.

Title: Unsupervised Open Information Extraction with Large Language Models

Author: Amrita Harikrishnan Nair

Institute: Institute of Formal and Applied Linguistics

Supervisors: doc. RNDr. Pecina Pavel, Ph.D., Faculty of Mathematics and Physics; and Prof. Dr. Günter Neumann, Faculty of Philosophy, Saarland University

Abstract: Open Information Extraction (OpenIE) is an NLP task that involves extracting entities and the relationship between them in an unsupervised, domain-agnostic manner. OpenIE has benefited from the use of Large Language Models (LLMs) and one such system that leverages the power of LLMs for OpenIE is DeepEx (Wang et al., 2022). DeepEx generates triples using the attention mechanism and then ranks them using a contrastive ranking model that uses 6.2M sentences in its training phase. The aim of this thesis is to study the effects of replacing the contrastive ranking model with a lighter, linguistic acceptability model that would rank the triples based on their acceptability. Upon analysis of different types of models that have been fine-tuned on varying amounts of data (relevant to the linguistic acceptability task), it is found that even the base `bert-large` model (not trained on any data) is capable of competing with DeepEx. Upon using a dataset of 8500 samples, the proposed system surpassed the DeepEx system by 0.5 points on the OIE2016 benchmark (Stanovsky, G. and Dagan, I., 2016). The performance on other evaluation benchmarks is equally impressive. A thorough analysis of the results, followed by ablation studies that provide insight into the performance of the proposed system, is performed, leading to new insights as to the quality of the proposed method.

Keywords: Open Information Extraction, Large Language Models, Linguistic Acceptability

Contents

Introduction

Natural Language Processing (NLP) is an interdisciplinary field that merges linguistics and mathematics, where the quantitative aspects are delivered through algorithmic thinking and computation, which enables machines to understand, interpret, and generate human language. The usage of NLP-inspired methods for the manipulation of data has proven pivotal in recent years, with methods grounded in linguistic theories gaining prominence. For instance, research into cross-lingual transfer across languages considers language similarity as a key feature for effective transfer, even demonstrated by Wu et al. [2019] that for multilingual-BERT [Devlin et al., 2018] and other multi-lingual models, transfer performance drops as the language pairs become linguistically distant.

NLP uses statistical, rule-based, and machine learning methods to deal with data. Since data is only as useful as the methods used to extract relevant information from it, without effective analytical and processing techniques, even large data sets can fail to provide meaningful insights. Towards this end, an important sub-field of the NLP space was formulated, which is, Information Extraction (IE). Succinctly defined by Wang et al. [2021], IE is the task of extracting *structured* information from *unstructured* sources. Effectively, this composes the information into a format that is ideal for querying, analyzing, and organizing data. Information in this structured format can then be used for tasks such as knowledge graph construction [Muhammad et al., 2020], knowledge base population [Angeli et al., 2015], etc.

The process of extracting relevant information from data is nuanced and not as straightforward as it may initially seem. Not all data is the same; hence, it cannot be treated uniformly. An intuitive way to present, store, and extract information from a piece of textual data can be to convert it into a set of n-ary tuples, such that each tuple (in the case of a ternary tuple) could be a separate `subject-predicate-object` combination.

Given two text entities in a sentence (which is part of a larger corpus), defining a set of relations/predicates that would fit each entity pair in the corpus seems like an impossible and imprudent task. Hence, there is a need for Open Information Extraction (OpenIE). As defined by Liu et al. [2022],

> *Open Information Extraction targets extracting structured information from unstructured text without limitations on the relation type or the domain of the text.*

The first known mention of OpenIE was by Yates et al. [2007]. The researchers mention a couple of key features that set OIE apart from other information extraction paradigms, which are **scalability**, **efficiency**, and the fact that it is a relatively **domain-agnostic automated** way of extracting relations and its entities. Over the last decade, significant advancements have been made in this area.

Large Language Models (LLMs) have demonstrated an innate understanding of language by capturing complex linguistic patterns and structures through extensive training on diverse, multilingual data. For example, Hewitt and Manning [2019] show that entire syntax trees are embedded in a language model's word representation space. LLMs have also proven their ability in various downstream tasks, most importantly, OpenIE, for this thesis.

Wang et al. [2021] describe DeepEx, an LLM-based method, as a translation framework, translating text to triples for three tasks, which are: OpenIE, relation classification, and factual probe. The system is divided into two steps: generation of the triples and ranking of the triples to extract top-k triples for evaluation. DeepEx professes state-of-the-art performance across datasets and tasks using its unique method of utilising the attention mechanism to choose relation phrases that are relevant for any two entities in the sentence. The analysis by Wang et al. [2021] suggests that it is *possible to transfer the inherent knowledge learned by a pre-trained LM to downstream tasks.* Wang et al. [2021] have graciously provided the code required to reproduce their results here [1]. The codebase makes it easy to extract the triples after the generation step and substitute the contrastive model used by DeepEx for ranking with other mechanisms.

Given a sentence, it is natural for humans to intuitively extract triples from it that strictly fit the `subject-predicate-object` pattern. For example, in the following sentence:

```
Mary had a little lamb and a goldfish
```

Two `subject-predicate-object` triples can be retrieved. The first is `Mary` - `had a` - `little lamb` and the second is `Mary` - `had a` - `goldfish`. Upon joining the tuple, due to the SVO word-order property of the English language [Tomlin, 1986], two coherent and grammatically correct/linguistically acceptable sentences are formed. Hence, a question arises:

> *To what extent does the syntactic accuracy and coherence of the sentence (formulated by joining the* ***`subject-predicate-object`*** *triple) correlate with the ranking of the retrieved triple (for the OpenIE task) and further the F1 score of the OpenIE system?*

"Linguistic acceptability" was a phrase first introduced by Noam Chomsky to *"...use the term acceptable to refer to utterances that are perfectly natural and immediately comprehensible without paper-and-pencil analysis, and in no way bizarre or outlandish,"* [Chomsky, 1965]. Since the 1960s, it has been a heavily researched topic, with a lot of the discussion bleeding into the areas of computational linguistics. The broader discussion around linguistic acceptability and the relationship between the acceptability of a sentence and its probability of occurrence (according to a Language Model (LM)) has been discussed in detail in Section 1.2.

Merging the areas of OpenIE and linguistic acceptability is the highlight of this document, which also covers the "syntactic" part of this thesis. In short,

[1] `https://github.com/wang-research-lab/deepex/`

triples for OpenIE will be obtained and then ranked according to their linguistic acceptability. The “semantic” part is covered by the use of SRL (Semantic Role Labelling) approach.

To summarize, the main contributions of this thesis consist of testing a hypothesis regarding the relation between the acceptability of a joined triple and its effect on the F1 score of the OpenIE task. Since LMs technically do “model languages” as spoken and written by humans, attempting to use LMs to extract entities and their corresponding relations in a way a human would is one of the primary aims of this thesis. Additionally, since linguistic acceptability covers a syntactic approach, a semantic approach using two entities and their connecting roles/relations for the task of OpenIE is another element of this thesis.

A number of prominent evaluation benchmarks, each from a different domain/evaluation method, are used. The proposed method performs as well as most other approaches for most of the evaluation datasets, surpassing DeepEx by a small margin for the OIE2016 dataset [Stanovsky and Dagan, 2016]. The performance on other datasets, such as PENN [Xu et al., 2013], is impressive. Compared to the ranking methodology used by DeepEx, the proposed method uses a dataset that is a fraction of the size of the dataset used to train the DeepEx ranker.

Dedicated to the open-source cause, this thesis is also focused on contributing to the DeepEx code-base, one of the contributions being the addition of evaluation using the CaRB [Bhardwaj et al., 2019] dataset. The pull request [2] is currently under review. The code for the approaches tried in this thesis is available here. [3]

[2] `https://github.com/wang-research-lab/deepex/pull/20`

[3] `https://github.com/amrtanair/master_thesis`

Chapter 1

Background

This chapter identifies the relevant literature and describes the concepts that are used in this document. The key ideas and debates in the fields of OpenIE and linguistic acceptability are presented to support the arguments made throughout the rest of the thesis.

This literature survey focuses on a number of topics, starting with a section on **Large Language Models (LLMs)**. The salient features of LLMs are described, especially the ones that are relevant to the content of this thesis. This is followed by a brief background on the main topic of this thesis, which is **Open Information Extraction (OpenIE)**. Furthermore, and most importantly, previous work on the central topic of this thesis, that is, systems that use ***LLMs for OpenIE*** are described.

The concept that is most referred to in this thesis, that is, apart from strictly computational Natural Language Processing (NLP) concepts, is **linguistic acceptability**. Section 1.2 elaborates on the meaning of the concept. Additionally, arguments directly relevant to the topic of this thesis are discussed, and the position taken by this thesis is stated and defended.

Section 1.3 details the concept of **Word Order** and its relevance to the core idea of the hypothesis that is central to this thesis.

Tying together these areas, the hypothesis of the thesis is stated in Sections 2.2 and 2.3. Further chapters are devoted to testing the hypothesis and analysing the results.

1.1 A short insight on Large Language Models

Shanahan [2024] simply define Large Language Models (LLMs) as *"...generative mathematical models of the statistical distribution of tokens in the vast public corpus of human-generated text, where the tokens in question include words, parts of words, or individual characters—including punctuation marks."*

LLMs have revolutionized the field of NLP by changing how machines understand and generate language. As is apparent in the name, these models are characterised by their large size [Zhao et al., 2023] and sophisticated architecture. LLMs are adept at utilizing deep learning architectures like **Transformers** (described in Section 1.1.2), which makes them capable of understanding the statistical properties of language, capturing nuances and complexities. This also makes them excel in dealing with various tasks like summarization [Jin et al., 2024], question answering [Yang et al., 2020], text generation [Topal et al., 2021], and text classification [Tezgider et al., 2022].

BERT [Devlin et al., 2018], GPT-2 [Radford et al., 2019], and T5 [Roberts et al., 2019] are LLMs that have been incredibly popular for their versatility in a variety of tasks. Each LLM is trained in a distinct manner, with a distinct combination of datasets. One of the best ways to leverage the power of LLMs is through **fine-tuning**. Radford et al. [2018] presented the concept of first *pre-training* a model on large amounts of general language data and the *fine-tuning* for specific tasks on their specific datasets. For example, text classification is a significant task in NLP and is also the task that is performed extensively in this thesis. Fields et al. [2024] mention numerous methods, datasets, and models to accomplish this task in various flavours, ranging from sentiment analysis and question answering to syntactic parsing.

Understanding the architecture of most LLMs requires two essential concepts: **Transformers** and the **attention mechanism**. The attention mechanism is described in Section 1.1.1, while the Transformer architecture is presented in 1.1.2.

1.1.1 Attention Mechanism

This concept, introduced by Bahdanau et al. [2014], is essential to popular neural network architectures and, more importantly, to the Transformer architecture. The core idea of the mechanism is that, for a particular sentence, certain other parts of the sentence are "more" important and need to be given "attention," which provides context for the name. Focusing on certain phrases that have already been generated while generating the next word enables better learning of relationships and dependencies within the data.

The attention mechanism seeks to combat the problem of fixed-length encoding performed by the encoders, which deprives the decoder of the opportunity to generate a sufficiently accurate result. The mechanism proposed allows the decoder to "look" at certain parts of the sentence that are relevant to predicting the target word. The authors Bahdanau et al. [2014] note that the parts of the sentences that the decoder attends to agree with their intuition.

The attention mechanism used by the transformer architecture has certain differences compared to the one introduced by Bahdanau et al. [2014]. The transformer uses a scaled dot-product attention mechanism, which calculates attention scores by calculating a weighted sum of values V, where the weights are determined by the similarity between queries Q and keys K. The formula for the attention mechanism is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Here, Q, K and V represent the query, key, and value matrices respectively.

The process of calculating the attention matrix involves the following steps. The input sequence is transformed into query, key, and value vectors using linear transformations. The similarity between the query and key vectors is then computed through dot products. To stabilize training, the dot products are scaled by dividing them by the square root of the key vector dimension. The scaled dot products are passed through a softmax function, producing attention weights. These weights determine the contribution of each value vector to the final output. The attention matrix is formed by the pairwise attention weights between the query and key vectors.

Next, the attention weights are applied to the value vectors using element-wise multiplication, and the results are summed. This step generates a context vector that captures relevant information based on the attention weights. The context vector is then utilized in subsequent layers for further processing or serves as the output for the task at hand[1].

By calculating the attention matrix, the Transformers architecture is able to capture dependencies and focus on significant segments of the input sequence.

The attention matrix is a key component of this thesis, as will be described in future chapters, especially the description of the DeepEx system [Wang et al., 2021] in Section 2.1.

1.1.2 Transformers

The **Transformer** architecture, introduced by Vaswani et al. [2017], is an integral part of most LLMs. It is quite efficient in processing sequential data in parallel through the attention mechanism.

This architecture is used chiefly for sequence-to-sequence tasks (shorthand: seq2seq). A short description of how an input sequence generates output is presented below:

Encoder phase: An encoder processes input sequences to transform its contextual and semantic information into a high-dimensional representation, that is, each word in an input sequence is converted into a high-dimensional embedding vector. The embedding vector for each input word is modified by adding it (element-wise) to a positional encoding vector of the same length, which introduces positional information. The vectors are passed to the encoder, which includes two sub-layers. The first sub-layer includes the self-attention mechanism. The second layer is a fully connected feed-forward neural network. The bidirectional nature of the encoder allows it to consider all words in the input sequence, regardless of their position relative to the word in focus.

Decoder phase: A decoder generates output sequences from encoded input representations, that is, it takes its own predicted output at the previous time step as input. Similar to the encoder, the decoder's input is modified with positional encoding. The augmented decoder input undergoes three sub-layers in the decoder block, with masking in the first sub-layer to prevent attention to subsequent words. In the second sub-layer, the decoder incorporates the encoder's output, enabling it to attend to all words in the input sequence. Next is a fully connected layer, after which a softmax layer [Bridle, 1989] is used to generate predictions for the next word in the output sequence.

The LLM used most prominently in this thesis is BERT (Bidirectional Encoder Representations from Transformers) [Devlin et al., 2018], explained in Section 1.1.3. BERT utilizes only the encoder, focusing on capturing the contextual relationships between words in a sentence through bidirectional attention, while the decoder is omitted.

[1] https://machinelearningmastery.com/the-attention-mechanism-from-scratch/

1.1.3 BERT

BERT is a Pre-Trained Language Model (PLM) developed by Google in 2018 [Devlin et al., 2018]. As described by the authors, BERT is *"...designed to pre-train deep bidirectional representations from the unlabeled text by joint conditioning on both left and right context in all layers"*. The most important feature of BERT is in its **bi-directionality**. This bi-directionality contrasts with earlier models that processed text in one direction, hence providing BERT with a more comprehensive understanding of language nuances and dependencies.

BERT is a Masked Language Model (MLM), specifically, it masks certain words in sentences and tasks the system with predicting those masked words based on their context within the sentence. It is pre-trained on a large amount of unlabelled data (English Wikipedia and Book Corpus [Zhu et al., 2015]).

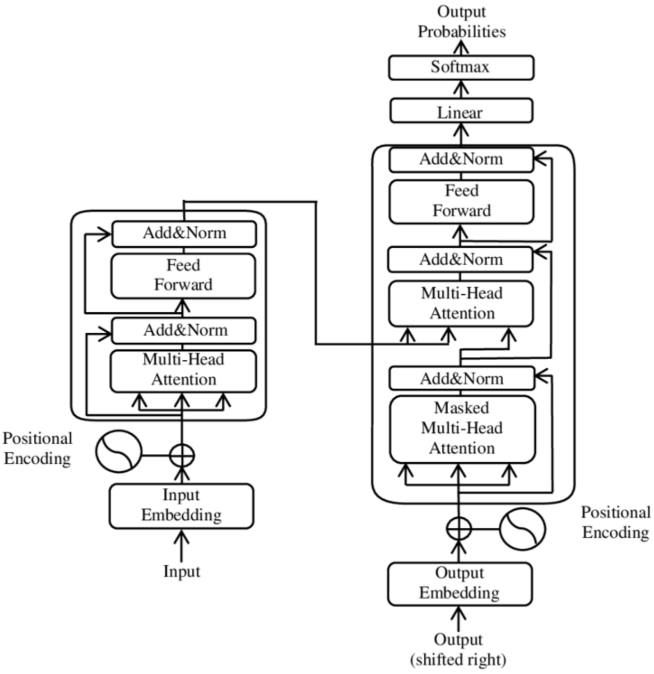


Figure 1.1: The Transformer architecture, as presented by Jia [2019]. The BERT architecture uses only the encoder component.

The BERT model is a very essential part of this thesis. First, the DeepEx model [Wang et al., 2021] uses it for the generation of the triples, as well as for the contrastive ranking model. Later, this thesis substitutes the contrastive ranking model with a **linguistic acceptability** model that ranks the triples on the basis of their acceptability for the OpenIE task. Further analysis is also performed on this model.

1.2 What is Linguistic Acceptability?

The proposed methodology in this thesis heavily depends on the concept of linguistic acceptability and its relation with `subject-verb-object` triples. The Linguistic Acceptability task is defined in many places as: *"...the task of deter-*

mining whether a sentence is grammatical or ungrammatical.” [2]. However, this is a very *computational* view of the situation. A linguistic view of the situation defines the situation of a sentence being classified as acceptable when *“...human subjects rate a sentence as acceptable”* [Lau et al., 2017]. This view does not consider the inherent “correctness” of a sentence, based on grammatical rules: it values only the opinions of human judgments. The grammatical correctness of a sentence *is* a factor in the judgment; however, it is not the only factor.

“Colourless green ideas sleep furiously” is a popular example of a sentence that is grammatically accurate but semantically does not make sense. This example was introduced by Noam Chomsky [1975].

The sentence is meaningless because ideas cannot sleep; neither can an idea be colourless, nor can it be green. However, syntactically, the sentence is grammatically correct according to the rules of the English language; it follows a subject-verb-adverb structure where “colorless” serves as an adjective modifying “ideas.”

This sentence demonstrates that the rules of syntax can be separate from the rules of semantics. Despite being grammatically correct, the sentence fails to communicate anything meaningful, hence proving a certain level of independence of syntax from semantics.

In this thesis, the distinction between linguistically acceptable and grammatically correct is particularly important for the following two reasons.

Firstly, the core idea of this thesis is to parrot the way a human would extract triples from a sentence. Intuitively, a human would extract triples from a sentence in the form of a clause that is *linguistically acceptable*. Secondly, in later chapters, due to the scarcity of data (for the linguistically acceptable task), *over-sampling* (a technique used to address class imbalance by increasing the number of instances of a particular class to achieve a more balanced dataset) was a method that was considered. One of the ways of implementing this would be to add noise to the existing linguistically acceptable samples by adding/swapping/removing words in the sentence, similar to the methods used by Feng et al. [2020]. This would potentially create grammatically incorrect sentences; however, their linguistic unacceptability would not be guaranteed unless a human would judge it so.

Hence, a clear distinction between these two concepts is necessary, which is provided in Section 1.2.1.

1.2.1 Linguistic Acceptability v/s Grammatical correctness

Lau et al. [2017] provide the following definitions.

Grammatically Correct

Grammaticality refers to the faithfulness of a sentence to the syntactic rules of a language. Grammaticality represents the *theoretical construct.*

[2] https://paperswithcode.com/task/linguistic-acceptability

Linguistically Acceptable

Linguistic Acceptability covers factors beyond grammaticality, focusing on semantic coherence, processing ease, and human judgment.

1.2.2 Ordinal versus binary acceptability judgments

Ordinal acceptability judgments refer to a sliding scale of acceptability where one end of the spectrum certifies the sentence as linguistically unacceptable, while the other declares the sentence as linguistically acceptable based on human judgments.

Binary acceptability judgments refer to a sentence falling into the acceptable or unacceptable class with no flexibility.

This thesis deals with two linguistic acceptability datasets. One is the Corpus of Linguistic Acceptability (CoLA) [Warstadt et al., 2019], and the other is the MegaAcceptability dataset [An and White, 2019]. The former contains binary acceptability judgments, while the latter contains ordinal acceptability judgments.

The literature regarding ordinal versus binary judgment leans either toward one direction or the other. The work done by Lau et al. [2017] leans towards acceptability being better represented in an ordinal format but does not discount the value of binary judgments. In their paper, An and White [2019] replace the ordinal judgments with binary judgments and then perform further analysis.

This thesis uses **binary acceptability** judgments for both datasets. This is because the CoLA dataset uses the same, and for parity's sake, the MegaAcceptability dataset was also adapted to reflect binary acceptability judgments.

1.3 Word Order in the English Language

As the phrase implies, word order is the pattern in which the syntactical constituents of a sentence generally appear in a language, as defined by Dryer [2007]. Further, the authors state that *"word order refers more generally to the order of any set of elements, either at the clause level or within phrases, such as the order of elements within a noun phrase."*

Primarily, word order in a language concerns itself with the position of the subject, verb, and object in a sentence. A loose definition of the terms subject, verb, and object is provided here, paraphrased from a text that deals with word order literature by Tomlin [1986].

Subject (S) *refers to the primary syntactic relation borne by an NP (noun phrase) with respect to the verb.*

Object (O) *refers to the secondary grammatical relation borne by an NP with respect to the verb.*

Verb (V) *refers to the verb root, and any bound morphemes may include tense, aspect, agreement markers, pronominal clitics, directional markers, and so on.*

The work in this thesis hinges on the premise that English is a `SVO` language [Tomlin, 1986]. Applying the proposed method to languages that do not form a strict word order or that do not follow the `SVO` word order might be challenging. This is identified as a limitation of the proposed method.

Chapter 2

Foundations of the Thesis

The previous chapter established the current research in the fields of OpenIE and linguistic acceptability. This chapter describes the basis of the thesis, identifying potential arguments for the primary conjecture of the thesis: *the linguistic acceptability of a triple may be used to rank it for the OpenIE task.* A rigorous description of the DeepEx framework (which provides the triples) is followed by a justification as to the reasons that this conjecture could be valid. Further, testing this conjecture involves bridging the gap between the OpenIE task and linguistic acceptability. Finally, a refined problem statement and the contributions of the thesis will be discussed.

2.1 About DeepEx

According to Wang et al. [2021], DeepEx utilizes a text-to-triple translation framework for information extraction tasks. In contrast to conventional methods relying on task-specific datasets and models, DeepEx approaches the task as a *translation between task-specific input text and output triples.* This design enables task-agnostic translation by leveraging the inherent knowledge within a pre-trained language model, enhancing adaptability and efficiency in managing diverse information extraction tasks.

The tasks DeepEx deals with are OpenIE, relation classification, and factual probe. It treats language models as zero-shot information extractors. Since LLMs seem to store relational information [Petroni et al., 2020], the inspiration for this paper was to leverage this relational knowledge. The authors utilize the power of the self-attention mechanism for this purpose.

2.1.1 Generation of triples

The architecture of this approach is shown in Figure 2.1. Noun phrases are extracted using Spacy [Honnibal et al., 2015–]. Each pair of noun phrases is treated as an argument pair for which the predicate needs to be extracted. The pairs are considered only in one direction at a time, but the algorithm is run in both directions since some triples are often in reverse order. The predicate is extracted using the combination of words in the sentence that give the highest **attention scores**.

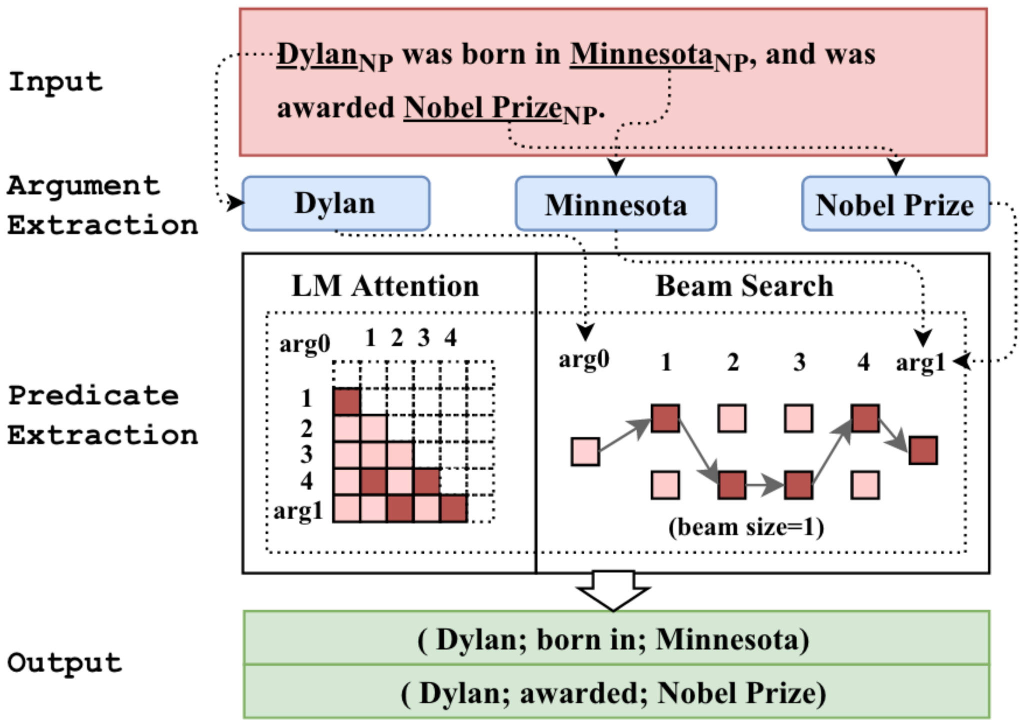


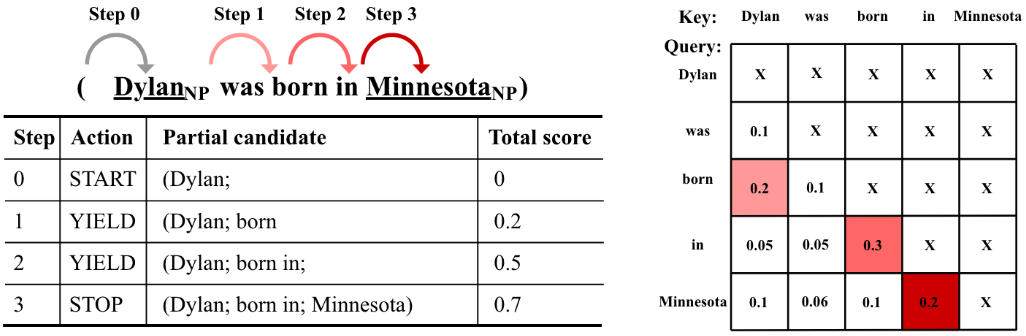
Figure 2.1: Wang et al. [2022]

An example is provided in figure 2.2. The example phrase `Dylan was born in Minnesota and was awarded Nobel Prize` is used where "Dylan," "Minnesota" and "Nobel Prize" are the noun phrases that have been extracted by Spacy.

Consider the sub-phrase `Dylan was born in Minnesota`. Here, "Dylan" and "Minnesota" are the noun phrases whose predicate has to be extracted. The right part of Figure 2.2 shows the attention matrix for this sub-phrase. Let "Dylan" be `ARG1` and "Minnesota" be `ARG2`. In the attention matrix, the word with the highest attention score while attending to `ARG1` is "born." This word is added to the predicate phrase, and the next step is to look at the attention scores for the word "born." The next word is "in," with an attention score of 0.3. Then, "in" is added to the predicate phrase, and the total score for this computation is increased by 0.3. The algorithm can be followed using the table that is on the right in Figure 2.2. The algorithm stops upon encountering `ARG2`, at which point the total score is 0.7.

Computing a score for every potential sequence proves to be computationally expensive, especially when dealing with long sequences. Consequently, the exhaustive search approach becomes awkward to manage due to the size of the sequences increasing as the sentences get longer.

To tackle this issue, the authors adopted **beam search** as an approximate strategy to efficiently explore the search space. Beam search operates by maintaining the k-best candidates, enabling more manageable computation and reducing the complexity of the search process. Using beam search, the process is



Step	Action	Partial candidate	Total score
0	START	(Dylan;	0
1	YIELD	(Dylan; born	0.2
2	YIELD	(Dylan; born in;	0.5
3	STOP	(Dylan; born in; Minnesota)	0.7

Query: \ Key:	Dylan	was	born	in	Minnesota
Dylan	X	X	X	X	X
was	0.1	X	X	X	X
born	0.2	0.1	X	X	X
in	0.05	0.05	0.3	X	X
Minnesota	0.1	0.06	0.1	0.2	X

Figure 2.2: Wang et al. [2022]

dependent on the beam size and not on the size of the sequence. This concludes the **generative process**.

2.1.2 Ranking of triples

In the generating stage, k candidate triples are generated for each argument pair. In the ranking stage, the task-specific relevance of triples is determined using a contrastive ranking model that is trained on a relational corpus that is not specific to the task at hand. However, it is crucial to note that the sequences within these candidates pertain not only to the relational aspect but also to the argument pairs themselves. *The primary objective of the ranking stage is to identify triples that explicitly convey the relational information between the argument pair.*

Method	F1	AUC
DeepEx	70.9	57.4
No ranking model	39.6	13.8

Table 2.1: From Wang et al. [2021]. Ablation of DEEPEX on OIE2016 dev set.

The model is trained on batches of N sentence-triple pairs, predicting which actually appeared among N^2 possibilities.

An example for the process would be as follows. Consider a sentence A with triples $A1$, $A2$ and $A3$, each triple being of the format subject-predicate-object. Another sentence B has triples $B1$, $B2$ and $B3$ in a similar format. The positive sentence-triple pairs would be formed among sentence A and its triples in the following manner:

```
[CLS] sentenceA [SEP] A1 [SEP]
[CLS] sentenceA [SEP] A2 [SEP]
[CLS] sentenceA [SEP] A3 [SEP]
```

Sentence B would also have three sentence-triple pairs. These would constitute the *positive* pairings.

For the negative pairings, we would combine sentence A with the triples of sentence B like so: `[CLS] sentenceA [SEP] B1 [SEP]`. Hence there would be 6 negative pairings too (for sentence A and sentence B).

The ranking system uses BERT as a base encoder, learning a shared space for sentence and triple embeddings in the `[CLS] sentence [SEP] triple [SEP]`

BERT format. *The goal is to maximize similarity for correct pairs and minimize it for incorrect ones, optimizing a cross-entropy loss.*

The authors of DeepEx train the system on `T-REx` [Elsahar et al., 2018], a dataset aligning Wikipedia abstracts with Wikidata triples.

The task-agnostic ranking model has a consistent input format for all tasks. During testing, input text and candidate triples are ranked using contrastive loss, and the top-n candidates become the output—e.g., top-2 triples for OIE.

The authors of DeepEx note that the ranker works very well, with nearly perfect top-1 accuracy. This finding is also supported by the results in Table 2.1. The triples from the generation stage account for around 39 F1 score points, while the addition of the ranking model significantly bumps up the score to 71 points. Hence, it can be concluded that the contrastive ranking model is crucial to the success of DeepEx.

By substituting the contrastive ranking model with a simpler linguistic acceptability model, this thesis attempts to replicate similar results as obtained by DeepEx using far fewer data points.

Another tentative hypothesis here is that, since the contrastive ranking model uses a large amount of data to minimize the distance between triples that *"actually appear in the sentence"* in the contrastive embedding space by optimizing a cross-entropy loss over the similarity scores, the triples that score higher are probably going to form cohesive and grammatically correct clauses, since these clauses *"actually appear in the sentence"* together. This is another hypothesis that can be substituted for the contrastive ranking model with the linguistic acceptability model. The next section justifies the substitution in more practical terms.

The ranking model has been published by the authors of DeepEx at the URL [1].

2.2 The conjunction of OpenIE and Linguistic Acceptability

OpenIE and linguistic acceptability seem like two disparate fields: one deals with extracting relevant information in an unsupervised manner. At the same time, the other addresses the vague and broad question of the quantity (on an ordinal scale)/class (in a binary interpretation) of linguistic acceptability a piece of text exhibits. However, an indirect relationship can be derived between the two.

ClauseIE [Del Corro and Gemulla, 2013] is a **clause-based** approach to OpenIE. The authors mention an example:

```
A. Einstein, who was born in Ulm, has won the Nobel Prize
```

This extraction of information, which has no predefined relations, is not just unsupervised, domain-agnostic, and scalable but also a low semantic representation of large amounts of natural language [Christensen et al., 2010]. The expected results from an OIE system would be:

[1] `https://huggingface.co/Magolor/deepex-ranking-model/`

```
A. Einstein, has won, the Nobel Prize
A. Einstein, was born in, Ulm
```

This extraction can be used to answer queries and other use cases and can also be considered as two clauses of the sentence.

It is remarkably similar to how a human would compile information from a given sentence, allowing these extractions to solve shallow semantic queries like `Who won the Nobel Prize?`.

ClauseIE utilizes the `SVO` (Subject-Verb-Object) word order property of the English language to recognise and extract clauses from sentences. Using dependency parsing and a domain-independent lexicon, the system detects clauses based on their grammatical structure, which involves identifying clauses that follow the SVO word order property.

A similar approach could be followed. Consider the two `NPs` extracted by Spacy to be a subject and an object and the relation between them, which is collected based on maximum attention scores as a possible verb phrase. Joining the triple would result in a clause [Langacker, 2015], which follows SVO word order property and, hence, would probably be a linguistically acceptable entity.

English follows SVO ordering and exhibits little flexibility [Downing and Noonan, 1995], [Tomlin, 1986]. A significant number of languages in the world follow this word-ordering. Further descriptions of the SVO word order property are provided in Section 1.3.

The cornerstone of this thesis is the idea that, in general, *English adheres to the SVO word order and hence simply joining the triples generated by DeepEx should typically generate a linguistically acceptable sentence.*

Now that there is a fair connection between these two fields, a question arises: What methods would be used to assess this relationship? Chapter 3 describes the different ways that this relationship could be explored by creating systems that rank triples based on their acceptability. The next section defines the objectives of this thesis and mentions the methods used to achieve them.

2.3 Problem Statement

One of the subtle aims of this thesis is to use simple architectures, methods and datasets to achieve the same results as larger and heavier models. The work done in this thesis is encapsulated in this section.

Objectives

1. Investigate the relationship between the linguistic acceptability of joined triples and the performance on the OpenIE task.

2. Evaluate the influence of augmenting the probability scores of the joined OpenIE triples with acceptability measures. The study includes the assessment of various model types, ranging from standard versions of large language models (LLMs) to versions that have been fine-tuned on datasets of varying sizes. Results from these fine-tuned models, trained with different amounts of data, will be documented and analyzed.

3. Evaluate the performance of the proposed system on diverse benchmarks, including NYT, OIE2016, PENN, WEB, and CaRB. Conduct studies and analysis to determine the specific contributions of linguistic acceptability measures to the overall performance in the OpenIE task.

Chapter 3

Linguistic Acceptability for OpenIE

This chapter describes the approaches used, elaborating on the different concepts, methods, and datasets used to achieve the results.

Section 3.1 introduces the various datasets used in this thesis, which includes datasets used for training and testing the fine-tuned linguistic acceptability model as well as datasets used for the evaluation of the OpenIE task. This section also includes a deep dive into the different datasets used, with a detailed analysis of their contents.

Section 3.2 lays out the connection between the probability of a sentence and its acceptability, also mentioning the changes needed to adapt the probability of a sentence to reflect estimate acceptability judgments.

Section 3.3 describes the experiments conducted in detail, with the deviation in standard approaches and the reasoning behind implementation decisions. The section also details the overall strategy of the system, describing the process from start to end.

3.1 About the datasets

Two datasets are used for fine-tuning the models for acceptability, the **Corpus of Linguistic Acceptability** (henceforth referred to as the CoLA dataset) [Warstadt et al., 2019] and the **MegaAcceptability** dataset [An and White, 2019]. Furthermore, five datasets will be used for evaluating the resulting OpenIE system, which are: **OIE2016** [Stanovsky and Dagan, 2016], **CaRB** [Bhardwaj et al., 2019], **NYT** [Riedel et al., 2010], **WEB** [Mesquita et al., 2013] and **PENN** [Xu et al., 2013]. Each dataset is chosen for certain characteristics, elaborated below.

3.1.1 Linguistic Acceptability Datasets

A detailed description and history of research in the area of linguistic acceptability is provided in Section 1.2. Choosing appropriate datasets for a fine-tuning task is of utmost priority as most machine learning models are affected by the data provided to them. Inaccurate, inappropriate, or inadequately pre-processed data would inevitably lead to a model that does not perform well on its intended task.

The principle of garbage in, garbage out (GIGO) is proven appropriate here. Hence, a detailed description of both datasets is presented in this section.

This section also presents a detailed analysis of the two datasets, comparing and contrasting the basic attributes of the datasets, including attributes that would influence their efficiency in being used to fine-tune LLMs, such as class imbalance. The associated code is available at the URL [1].

The CoLA dataset

The CoLA dataset is part of the GLUE (General Language Understanding Evaluation) benchmark [Wang et al., 2018], which was created to encourage the development of models that would be able to complete a range of tasks, such as sentiment analysis and question answering across a range of domains. Wang et al. [2018] specify that the datasets included in the benchmark were chosen *"...because they have been implicitly agreed upon by the NLP community as challenging and interesting."*

The GLUE benchmark uses the CoLA dataset as the standard for linguistic acceptability, along with the Matthews Correlation Coefficient MCC) being specified as the metric to be used to compare linguistic acceptability models. A longer description of the metric is provided in Section 3.3.1.

Label	Sentence
1	The dancing chorus line of elephants broke my television set.
1	Gilgamesh is not reading the cuneiform tablets.
0	the logs piled the barge high.
1	Bill alleged that Roger had eaten the cake.
0	They can happy.

Table 3.1: A snapshot of the CoLA dataset

One of the primary reasons for choosing the CoLA dataset for these sets of experiments is that it is widely regarded as one of the most prominent and reliable datasets for assessing linguistic acceptability. The dataset is limited in its size: it contains around 10,500 sentences (8,500 for the train-spilt) from 23 different linguistic publications annotated for linguistic acceptability by its authors.

The dataset is fairly diverse, topics include negative polarity, verb alternations, dative alternations, comparatives and so on. Considering the limited size of the CoLA dataset, it would be interesting to compare the ranking performance (on the downstream OpenIE task) of an acceptability model trained on the CoLA dataset against one that is trained on a larger dataset.

Dataset Analysis

Table 3.1.1 presents an initial, rudimentary analysis of the data, specifically the train-spilt of the dataset. It is fairly obvious that the dataset skews towards linguistically acceptable sentences. This imbalance is significant because models that are trained on such skewed data without any class imbalance mitigation

[1] `https://github.com/amrtanair/master_thesis`

Table 3.2: Basic Statistics of the CoLA dataset

Statistic	**Value**
Number of Sentences	8,551
Shortest Sentence Length	2
Longest Sentence Length	42
Average Sentence Length	7.70
Median	7.0
Number of linguistically acceptable sentences	6,023
Number of linguistically unacceptable sentences	2,528

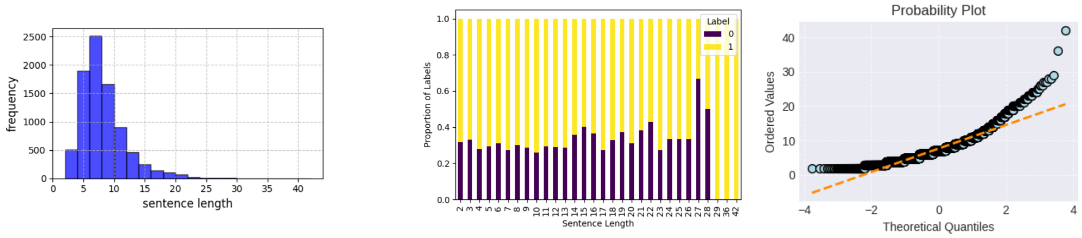


Figure 3.1: Statistical analysis of the CoLA dataset. From left to right, (a) Frequency vs. Sentence Length for the CoLA dataset, (b) Distribution of labels across sentence lengths, (c) Q-Q Plot of Sentence Lengths

techniques are likely to develop a bias toward labeling sentences as acceptable. Addressing this issue is essential for building an accurate classifier model.

The sentence lengths vary between 2 and 42, with the median being around 7 words. Figure 3.1(a) shows the variation in sentence length and the frequency associated with each sentence length interval. Figure 3.1(b) shows the proportion of labels against all values of sentence lengths.

The Shapiro-Wilk test [Shapiro and Wilk, 1965] for assessing the normality of distribution provided a statistic of 0.887 and a p-value of 0.0, which indicates a significant deviation from normal distribution in the sentence lengths. This shows that the sentence lengths do not follow a normal distribution. Since the Shapiro-Wilk test is unreliable for larger datasets (`>5000` data points) [Field, 2009], this result is supported by the Q-Q plot, 3.1(c).

To assess differences in sentence lengths between the acceptable correct and unacceptable samples, the Mann-Whitney U test [Mann and Whitney, 1947] was performed, resulting in a U-statistic of 7629584.5 and a p-value of 0.873. The high p-value suggests that there is no statistically significant difference in the distributions of sentence lengths between grammatically correct and incorrect samples. Thus, despite the non-normality of the sentence lengths, they do not significantly differ across the labels.

Another aspect of the dataset pertains to the type of grammatical errors present in the unacceptable sentences. A basic understanding of the errors would provide insight as to the kind of linguistic acceptability the model will be exposed to and will help in the downstream OpenIE task. Using the `language-check` library LanguageTooler [n.d.], the errors analysed are listed in table 3.3

Error(no. of occurrences)	Example
`AGREEMENT_SENT_START`(10)	The men doctors of medicine.
`MD_BASEFORM`(13)	He has will seeing his children.
`TWO_CONNECTED_MODAL_VERBS`(6)	Kim must will bake a cake.

Table 3.3: Snapshot of errors in the CoLA dataset (from LanguageTooler [n.d.]). The description of the errors can be found at `https://community.languagetool.org/rule/list`.

The MegaAcceptability Dataset

The MegaAcceptability dataset was first introduced by White and Rawlins [2016] as a way to address the task of studying the effect of a word's semantic characteristics on its syntactic usage pattern. Another reason for creating this dataset was to study the acceptability judgments of clause-embedding verbs.

Towards this end, a dataset was curated, which contained 50,000 sentences constructed by taking 1,000 verbs and 50 different syntactic frames. Each sentence contains a *acceptability judgment* between 1-7. The judgments were provided by native English speakers who rated the sentence on the basis of how "natural" or "acceptable" each syntactic construction was perceived.

Later, An and White [2019] extended the dataset to enhance its scope to understand and include the range of variability in *neg-raising inferences* across different contexts. Negation-raising inferences can be simply understood as the phenomenon where a negation applied to a verb can imply an application of the negation on the subordinate clause as well. Hence, a large number of acceptability judgments on negation-raising inferences were collected for English verbs that could embed clauses.

Version 1 of the MegaAcceptability dataset (`MegaAcceptability-v1`) contained sentences only in the past tense. Since the variability in negation-raising inferences is influenced by the tense of the clause-taking verb, the second version (`MegaAcceptability-v2`) modified some sentences by changing their verbs to the present tense to include examples of both tenses in the dataset. Any mentions of the MegaAcceptability dataset in this thesis refer to `MegaAcceptability-v2`.

The authors of MegaAcceptability chose sentences that were rated 4 (out of 7) or higher for their experiment. Similarly, for the purposes of this thesis, all sentences that are rated as less than or equal to 4 are tagged as `0` while sentences above this threshold are rated as `1`. The arguments for/against binary acceptability judgments are clarified in section 1.2.2. Further, each sentence was originally annotated for acceptability by a number of annotators. The judgments were averaged and then classified as `1` or `0` if they were `>4` or `<=4`, respectively.

Dataset Analysis

One of the primary reasons for choosing this dataset is its size, but also because of the linguistic information included in the dataset, such as the syntactic frame of the sentence and the verb around which the sentence is centered. The basic statistics of the dataset are presented in table 3.4

Table 3.4: Basic Statistics of the MegaAcceptability dataset

Statistic	**Value**
Number of Sentences	74,827
Shortest Sentence Length	2
Longest Sentence Length	10
Average Sentence Length	5.63
Median Sentence Length	6.0
Number of linguistically acceptable sentences	30,646
Number of linguistically unacceptable sentences	44,181

Exploring the nature of the syntactic information can be crucial in the analysis of the performance of the linguistic acceptability task, which further affects the downstream OpenIE task's results, too. Figure 3.1.1 describes the distribution of the syntactic frames across the dataset.

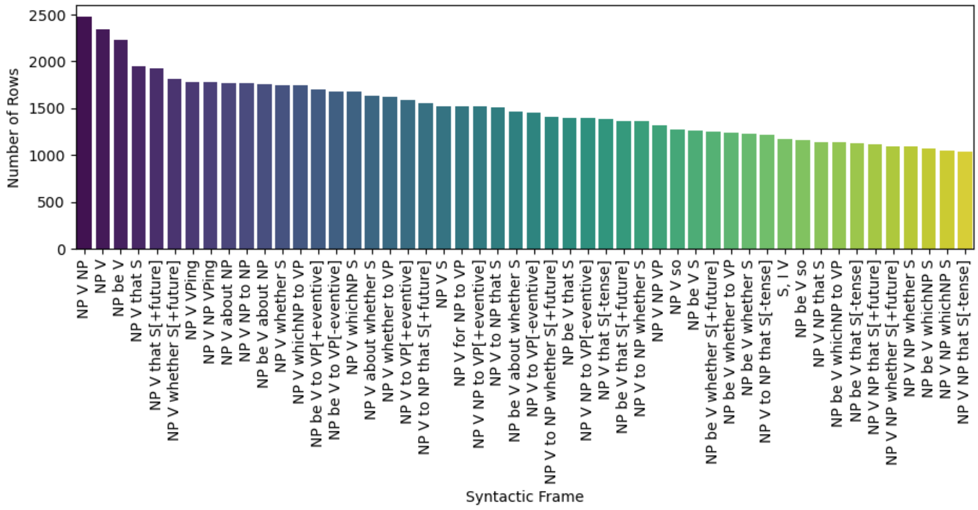


Figure 3.2: Number of samples for each syntactic frame

The `NP V NP` syntactic frame has the highest number of rows, followed by the `NP V` frame. An example, from the dataset, of the former syntactic frame is **Someone abhorred something**, while the latter frame is **Someone abhorred**.

One potential drawback of using this dataset is that, during its creation, the focus was entirely on the verbs, and hence, paraphrasing from An and White [2019], *"...the lexical content was kept minimal to avoid typicality effects, ensuring that the acceptability judgments reflect the syntactic frames and the tested verbs rather than the influence of specific lexical items."*

Figure 3.1.1 shows the mean response and the standard deviation for the top 5 and the bottom 5 syntactic frames. The frames here are sorted by their count. An elaborated version of the same is presented in Figure A.1. Having the syntactic frame `NP V NP` identified and present so prominently in the dataset is particularly beneficial for our use case since the DeepEx system extracts two `NPs` and then tries to use an attention-based mechanism to extract the relation between them, which

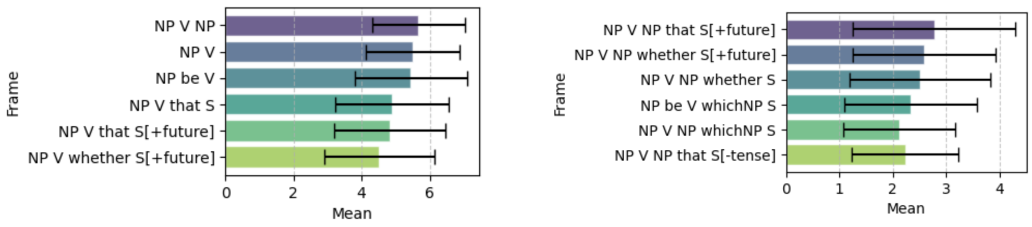


Figure 3.3: Distribution of acceptability judgments across syntactic frames. From left to right, (a) Top 5 frames, (b) Bottom 5 frames

fits neatly into the `NP V NP` frame. Using this dataset to fine-tune the linguistic acceptability model would present the model with positive and negative examples for this particular syntactic frame, which may improve the performance on the downstream OpenIE task. Further information about the dataset can be found in the Appendix.

3.1.2 OpenIE datasets

The following evaluation benchmarks were chosen for one of two reasons: either the benchmark was used by the authors of DeepEx to evaluate the system, or the dataset/evaluation script brings forward a new facet upon which the extractions could be judged. It is essential to understand the motivations behind the creation of each motivational benchmark so that the scores produced make sense in the context of the proposed system.

Dataset	Domain	#Sents
OIE2016	News, Wiki	3,200
WEB	News, Web	500
NYT	News, Wiki	222
PENN	Mixed	100

Table 3.5: Statistics for datasets. The dataset for the CaRB benchmark is the same as OIE2016. Source: Wang et al. [2021]

OIE2016

Published in 2016, this benchmark was the first labelled corpus created for the evaluation of OpenIE systems [Stanovsky et al., 2018b]. It is an automatically generated corpus; it uses the **QA-SRL** corpus [He et al., 2015], translating the samples to OpenIE-relevant samples. The authors of the benchmark clearly defined the guidelines they consider appropriate for the definition of an OpenIE task, which are ***assertedness*** *((Sam; succeeded in convincing; John) versus (Sam; convinced; John))*, ***minimal propositions****(condensing a sentence into small meaningful chunks that still convey all the information)* and ***completeness and open lexicon***.

PENN

This dataset was created by Xu et al. [2013] because they noticed a gap in the literature, where datasets which included *"all kinds of relations"* were scarce. This dataset for OpenIE was created using the PENN treebank [Marcus et al., 1993]. The paper whose by-product this dataset focused on relations and proposed two sub-tasks to the concept of OpenIE, the first being to check if a given sentence contains a relation between two entities and the second was to *"...confirm explicit relation words for two entities"*.

NYT and WEB

Mesquita et al. [2013] created the WEB and NYT datasets using WEB-500 and NYT-500. WEB-500 was developed by Banko and Etzioni [2008] and the authors mention that the WEB-500 data is *"...often incomplete and grammatically unsound, representing the challenges of dealing with web text"* while NYT-500 has *"...formal, well written new stories from the New York Times Corpus"*, [Sandhaus, 2008].

Mesquita et al. [2013] manually annotate the datasets by marking two entities, as well as defining a *span* of tokens that encompass relevant modifiers and connecting prepositions for each sentence. The dataset is collected from the Stanovsky et al. [2018a] repository.

CaRB

This evaluation benchmark did not introduce a new dataset, it simply refactors the OIE2016 dataset in a couple of different ways. The dataset is re-annotated using human judgments, and the scorer is updated to rectify certain drawbacks the authors of Bhardwaj et al. [2019] noticed in the OIE2016 scorer. This evaluation provides a contrast to the datasets above as they all used the same scorer as the one used by Stanovsky and Dagan [2016] even though the datasets are from different domains. For CaRB, however, the scorer is different, but the test sentences are the same as the one used by OIE2016.

The authors of CaRB note that OIE2016 has been known not to penalize long extractions or misidentifications of relations in argument slots, favoring systems that may produce trivial but high-scoring outputs. It also allows multiple occurrences of the same word in an extraction, which may result in a higher precision score.

These drawbacks of OIE2016 prompted the creation of the CaRB benchmark. The CaRB scorer use *structured tuple matching* and *multi-match recall*, allowing gold tuples to match multiple extractions while penalizing redundancy. Structured tuple matching aims to evaluate the semantic coherence and structural integrity of extracted information.

Chapter 4 presents the results across all benchmarks, which bring to light the various idiosyncrasies of each dataset and benchmark, highlighting the elements that the proposed system performs better or worse with.

3.2 Acceptability and Probability

Section 2.2 lays out the conjecture that the linguistic acceptability of a triple could be related to the OpenIE task. Assessing the linguistic acceptability of a triple is the main task of this section. From here on, linguistic acceptability will be referred to as acceptability for the sake of brevity.

An LLM is trained in an unsupervised manner on a large amount of text, with the model containing millions or even billions of parameters [Zhao et al., 2023]. In some situations, the corpus used to train the LLM is basically a large portion of the internet (for example, Common Crawl [Crawl]). The corpus contains, in general, linguistically acceptable sentences from various sources, such as Wikipedia, books, Reddit forums, and so on. Hence, it would be reasonable to assume that the probability of a sentence would provide some measure of its acceptability. However, the situation is not as straightforward. The authors Lau et al. [2017] put forward a couple of arguments to discuss the statement that the probability of a sentence according to a language model is not an exact comparison to its acceptability.

- The sum across all possible sentences for a probabilistic model would be 1, which would result in very low probabilities for most sentences, disregarding any notion of acceptability.
- If the probability of a sentence according to a language model would be considered similar to its acceptability, sentences with rare words would rate far lower than sentences with common words despite no apparent difference in the acceptability of the two. Lau et al. [2017] mention an easy example of this situation: the two sentences `I saw a yak` and `I saw a cat` should ideally not differ in acceptability, however, since `yak` is a much rarer word, the assigned probability for the first sentence would be lower than that for the second sentence.
- The factors that could potentially influence probabilities, such as sentence length and the probability of the constituent words, are different from the ones that determine acceptability. A sentence would be acceptable irrespective of its length or the uniqueness of its constituent words while these factors would influence probability. This highlights that the probability and acceptability of a sentence are measured in distinct ways.

However, these factors do not completely discount the role of probability for acceptability judgments. Due to the sheer quantity of data the LLMs are trained on, it is definitely possible that acceptable sentences are more likely to be assigned higher probabilities. Hence, it can be said that the relationship between probability and acceptability of a sentence is not direct but is influenced by other factors.

3.2.1 Acceptability Measures

Lau et al. [2017] introduce the concept of `Acceptability Measures`, which are basically normalisations/augmentations to the probability of a sentence according to a language model that would attempt to reflect its acceptability. Applying these normalization techniques to the probability of a sentence according to a language model may equate to its acceptability judgment. This section elaborates on the different acceptability measures employed in this thesis.

Log Probability (`LogProb`)

$$\text{LogProb} = \log p_m(s)$$

Calculates the logarithm of the probability $p_m(s)$ assigned to the sentence s by the language model m. The language model here can be the base model provided by HuggingFace [Wolf et al., 2019] or a fine-tuned linguistic acceptability model.

Mean Log Probability (`Mean LP`)

$$\text{Mean LP} = \frac{\log p_m(s)}{|s|}$$

Normalizes the probability of the sentence s w.r.t its length. Here, $|s|$ represents the length of sentence s, where the length is the number of characters in the sentence.

Normalized Log Probability (Division) (`Norm LP(Div)`)

$$\text{Norm LP (Div)} = \frac{\log p_m(s)}{\log p_u(s)}$$

Calculates the ratio of the log of the probability of the sentence s according to the model m to the log of the unigram probability of the sentence s. In the case of fine-tuned models, the numerator is the log of the probability assigned by the model that the sentence is linguistically acceptable. The denominator is the unigram probability of the model upon which fine-tuning has been conducted.

Normalized Log Probability (Subtraction) (`Norm LP (Sub)`)

$$\text{Norm LP (Sub)} = \log p_m(s) - \log p_u(s) = \log \frac{p_m(s)}{p_u(s)}$$

Computes the difference between the log probability of the sentence s under the model m and the log of the unigram probability of the sentence s. The notations used here as similar to the ones used in `Norm LP (Div)`.

Sentence Log Odds Ratio (`SLOR`)

$$\text{SLOR} = \frac{\log p_m(s) - \log p_u(s)}{|s|}$$

This measure, introduced by Pauls and Klein [2012] calculates the log odds ratio, which attempts to adjust for the frequency of individual tokens and provides a metric that can approximate *fluency*. Kann et al. [2018] describe SLOR as a *"...a normalized language model score, as a metric for referenceless fluency evaluation of natural language generation output at the sentence level."*

3.3 Experiments

The experiments apply **sentence-level** acceptability measures to the probabilities obtained from different models. Applying **word-level** measures, as defined by Lau et al. [2017], could present a relevant and potentially insightful approach for further exploration.

However, given the extensive scope of experiments (evaluation of five OpenIE benchmarks across multiple models, each producing results for five normalization schemes in addition to a non-trivial analysis of the results) conducted in this thesis, adding word-level measures would significantly exceed the practical limits of this study.

Focusing on selected methods would help maintain a manageable quantity of experiments. Therefore, experiments around word-level measures are left for future work.

The experiments are divided into two modes: the type of model used and the amount of data used to fine-tune the models. This approach demonstrates the impact different types of language models and dataset sizes have on the linguistic acceptability task, which may translate to an impact on the task of ranking OpenIE triples.

Three variants of `bert-large` will be used. The first being a basic `bert-large` model, both `cased` and `uncased`. The second is the `bert-large` model (`uncased` and `cased`) fine-tuned on the CoLA dataset. Finally, the third is the `bert-large` models (`uncased` and `cased`) fine-tuned on the MegaAcceptability dataset, which brings the total number of models to six. Hence, we end up with the following groups of experiments:

- **Group A**:
  1. The basic `bert-large-uncased` model,
  2. The basic `bert-large-cased` model,
- **Group B**:
  1. The `bert-large-uncased` model fine-tuned on the **CoLA** dataset ,
  2. The `bert-large-cased` model fine-tuned on the **CoLA** dataset.
- **Group C**:
  1. The `bert-large-uncased` model fine-tuned on the **MegaAcceptability** dataset,
  2. The `bert-large-cased` model fine-tuned on the **MegaAcceptability** dataset.

The results of each of these models will be passed through the five normalization methods and then evaluated on each of the five evaluation datasets mentioned in section 3.1.2.

A natural question would be the usage `bert-large-*` and not the base versions of the model (`bert-base-*`). This is simply because it was easier to obtain unigram probabilities for the larger model due to the work done by Lau et al. [2020] in the paper ***How furiously can colorless green ideas sleep? sentence acceptability in context***. The code for the same is available at the URL[2].

Obtaining unigram probabilities for the base models either through a literature survey or by building the dataset is earmarked as future work.

Two versions of the `bert-large` model are chosen, the `cased` and the `uncased` version. The reasoning behind this decision is that the impact of casing on acceptability is non-trivial. The entities named `apple` and `Apple` could signify two different concepts: the first refers to the fruit, and the second could either be the first word in a sentence about the fruit (hence, capitalized) or it could be a reference to the tech company.

One of the reasons for choosing `BERT` as the model that drives the methodology of this thesis was due to the constraint imposed by the availability of the unigram probabilities. This still leaves two other models that could be considered, the `GPT2-medium` and `XLNet`. Choosing `BERT` here was purely due to parity with the original DeepEx system as the generation of triples was through the `BERT-large` model while the ranking system was built using the `BERT-base` model.

Another reason was due to feasibility: running exhaustive fine-tuning on a variety of models is computationally expensive and tedious. Though Lau et al. [2020] have provided unigram probabilities for other models, due to the cost of running such computations and the manageability of diverse types of LLMs, only the `bert-large-*` models have been used for experiments. However, results for the `GPT2-medium` model has been reported, using acceptability measures, for the OIE2016 benchmark, in an effort to demonstrate the results obtained when not using a masked language model like BERT.

3.3.1 General Approach

Each experiment follows a similar strategy, with slight changes to reflect the differences in the experiments.

The intention behind the first two experiments (`bert-large-uncased` and `bert-large-cased`) is to set up a baseline. The computation of baseline results is essential to the task since they set up a foundation upon which further experimentation can be compared. Any improvements in the results can then be attributed to the changes made to the system and not due to inherent differences.

The next two experiments demonstrate the effect of fine-tuning each variant of `bert-large` on the acceptability task (with CoLA dataset) and then using the logits (the probability of being linguistically acceptable) obtained to rank the OpenIE triples. The logits are transformed into *acceptability measures* by using the normalization scheme described in section 3.2.1. The reasoning behind fine-tuning here would be to test the impact of providing the model with clear examples of linguistic acceptability and then assessing the results on the OpenIE task.

Finally, the last two experiments demonstrate the usage of a larger dataset for fine-tuning the linguistic acceptability model. MegaAcceptability is nearly

[2] `https://github.com/jhlau/acceptability-prediction-in-context/tree/master/code`

10 times larger than the CoLA dataset. The hope here is that fine-tuning on a larger dataset would result in a higher score for the acceptability task, which would, in turn, translate to a higher score for the downstream OpenIE task.

This sequence of experiments attempts to find a relation (if one exists) between the linguistic acceptability of a triple and its ranking in the OpenIE task. The overall strategy is as follows:

- Step 1, **Setup**: Prepare the **DeepEx** system by installing all relevant libraries and frameworks. Download evaluation datasets and choose a beam-size. If fine tuning a linguistic acceptability model, collect the dataset and the LLM.
- Step 2, **Generate triples**: Run the DeepEx system (until the generation step) on the chosen evaluation dataset.
- Step 3, **Training the linguistic acceptability classification model**: This step is skipped for the first two experiments. For the rest, based on the dataset and LLM combination, fine tune the model through hyperparameter fine tuning to achieve the best evaluation score possible for the linguistic acceptability task. The evaluation metrics used for the task are adapted from the GLUE benchmark.
- Step 4, **Running inference on the fine-tuned model**: Pass all triples for each sentence as input to the fine-tuned model. Depending on the experiment type, either the probability of the triple is used for calculations regarding the normalization methods or the *logits obtained from the linguistic acceptability model, which signify linguistic acceptability* are used.
- Step 5, **Evaluate the results** Run the evaluation script for each evaluation benchmark, for each model, fine-tuning dataset and normalization method combination. Record and compare the results.

Group A: Experiments on the `bert-large-cased` model and `bert-large-uncased` model

For every experiment, the most important value that needs to be calculated is the probability of the sentence according to the language model. Calculating the probability for a sentence is especially challenging for the `BERT` model since it is designed for masked language modelling and hence is not available, out-of-the-box, for probability computations. Additionally, and more importantly, the **bi-directional** nature of `BERT` makes probability computations tricky, since it considers both left and right directions simultaneously.

The code provided by Lau et al. [2020] at the following URL[3] was adapted to calculate the probability of each triple accurately. The code masks each token in the input sentence, one at a time, allowing the model to predict the masked token based on the context from both directions, and then sums up the log-probabilities of the correct predictions. For each normalization measure, evaluation dataset

[3] `https://github.com/jhlau/acceptability-prediction-in-context/blob/master/code/compute_model_score.py`

and model, the evaluation files were created and the results are presented in Table 4.2 and Table 4.3.

This sets the baseline upon which the results for the OpenIE task should ideally improve.

Group B: Experiments on the `bert-large-cased` model and `bert-large-uncased` model fine-tuned on CoLA

`BERT` has been one of the most popular LLMs in the NLP space, and its performance on the text classification task is one of its strengths [González-Carvajal and Garrido-Merchán, 2020], [Garg and Ramakrishnan, 2020]. The decision to fine-tune the BERT model on the CoLA dataset and not use the models available on the HuggingFace hub was due to a few factors, the foremost being control over the fine-tuning process. To demonstrate and validate the conclusions drawn in this thesis, it was necessary to have investigated at least one model(in this case, `BERT`) thoroughly and have no confounding/unknown factors regarding the training procedure associated with the analysis.

Another core reason for choosing to fine-tune and not use a ready-made model is the freedom to analyse the model by taking it apart to perform studies, as has been demonstrated in table 4.10.

Finally, experimenting on the differences between using a `cased` versus an `uncased` model is interesting from the point of view of the linguistic acceptability task due to the differences in tokenization, as described in section 3.3. `GPT2-medium` models are available only in the cased format[4] and further, the availability of unigram probabilities for both `cased` and an `uncased` model drove this decision.

The fine-tuning procedure was conducted using **bayesian search**, the starting hyper-parameters being the values mentioned in the BERT [Devlin et al., 2018] paper, which are:

- Batch size: 16, 32
- Learning rate (Adam): 5e-5, 3e-5, 2e-5
- Number of epochs: 2, 3, 4

Slight deviations were made from the above hyper-parameters. The batch-size recommendations were by-and-large maintained, however, another optimizer was added to the search space, including Adam [Kingma and Ba, 2014], which is, AdamW [Loshchilov and Hutter, 2017]. The learning rate values were maintained.

Early in the experimentation, the loss function supplied by the `BertForSequenceClassification` for binary classification is cross entropy loss[5] was used. However, it was quickly apparent that, thought the training loss reduced, validation and training accuracy increased as the training progressed, the validation loss ballooned to very high values. This could perhaps be due to the **class imbalance** in the CoLA dataset, as demonstrated in the data analysis of

[4] `https://github.com/huggingface/transformers/issues/2314#issuecomment-571059380`

[5] `https://github.com/huggingface/transformers/blob/9aeacb58bab321bc21c24bbdf7a24efdccb1d426src/transformers/modeling_bert.py#L1354`

the CoLA dataset in section 3.1.1, which could cause over-fitting and result in such training behaviour.

To rectify this, several techniques were considered, two of them being **under-sampling** and **oversampling**. Under-sampling was dismissed as a viable technique due to the scarcity of the linguistically unacceptable sentences. Under-sampling would have halved the dataset.

Oversampling would involve generating linguistically unacceptable (negative) samples. Initially, this approach was seriously considered, however, the reasoning by Lau et al. [2017] regarding the differences between grammatically unacceptable sentences being linguistically acceptable (described in section 1.2.1) holds true and hence, instead of modifying the dataset, other class imbalance methods were considered.

One such method was changing the loss function. Focal loss function, introduced by Lin et al. [2017] is specially designed to combat class imbalance. Replacing the cross entropy loss with focal loss stabilised the training process, and the acceptability task produced better results.

Another measure taken against over-fitting would be early stopping, specifically, early stopping based on validation loss. Hence, no epochs were defined in the search space, instead, training would run until the validation loss did not decrease significantly, or if the validation loss remained unchanged. Experimentation with a penalty mechanism was also implemented, such that, a penalty was applied when the validation loss did not decrease from one epoch to the next, but training was stopped only when the number of penalties rose to 2.

Finally, dropout in the model configuration and weight decay in the optimizer was set between the large interval of the default value (`0.1` for both) to 0.3, to provide a large space for the Bayesian search.

Five-fold cross-validation technique was employed to evaluate all models, ensuring robust performance scores. Each model was also trained and tested using 5 different random seeds to account for variability in initialization and training processes.

A fixed number was not set for the number of runs conducted using Bayesian search, primarily due to computational restraints. The search was stopped once there was a significant amount of runs (at least 10) and until there was only a minute improvement in the MCC (Section 3.3.1).

Finally, the following hyper-parameters were chosen:

Model	Batch Size	Epochs	Learning Rate	Optimizer	MCC	Accuracy
`uncased`	16	5	5e-05	AdamW	53.49	81.01
`cased`	16	3	5e-05	AdamW	56.11	81.97

Table 3.6: Fine-tuning the `bert-large` models on the CoLA dataset.

Group C: Experiments on the `bert-large-cased` model and `bert-large-uncased` model fine-tuned on MegaAcceptability

Similar training techniques as described for Group B were used for the MegaAcceptability dataset as well. The results of the experiment are presented below:

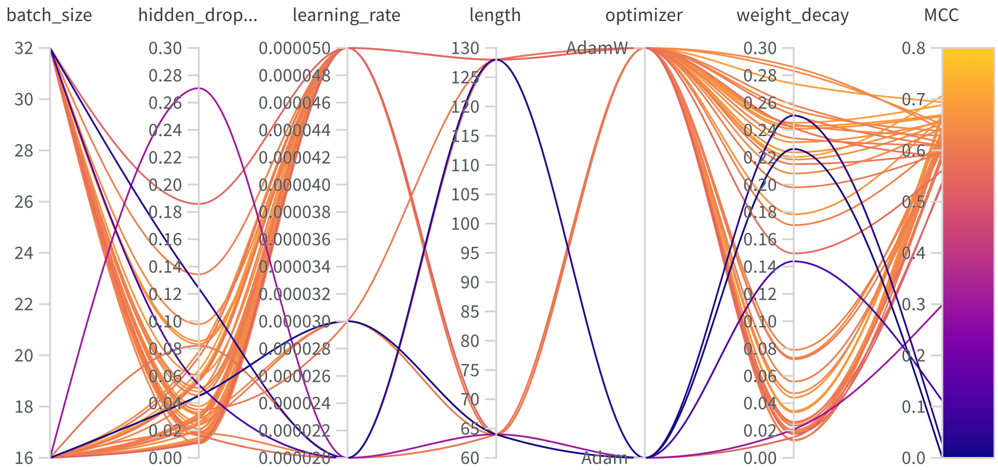


Figure 3.4: Bayesian search visualised for the `CoLA-BERT-large-uncased` model

Model	Batch Size	Epochs	Learning Rate	Optimizer	MCC	Accuracy
`uncased`	32	3	3e-05	AdamW	67.30	83.76
`cased`	32	3	2e-05	AdamW	68.51	84.13

Table 3.7: Fine-tuning the `bert-large` models on the MegaAcceptability dataset.

Evaluation Metric: Matthews correlation coefficient

The metric most often used for a variety of tasks would be the accuracy or the F1 score. However, for a binary text classification task, and specifically according to the GLUE benchmark, Matthews correlation coefficient (MCC), Matthews [1975] is used.

The coefficient takes into account both (true and false) positives and negatives while the F1 score ignores the true negatives in its calculation. Equation (3.1) shows the formula for MCC which can be compared against equation (3.2).

$$\mathrm{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.1}$$

$$F_1 = \frac{2\mathrm{tp}}{2\mathrm{tp} + \mathrm{fp} + \mathrm{fn}} \tag{3.2}$$

The MCC coefficient provides a value between `-1` and `1`. The MCC score and accuracy will be taken into account while evaluating the fine-tuned model.

3.3.2 Running inference on the fine-tuned models

First, a short definition of the evaluation metrics used for the OpenIE task is defined below. Next, the method of inference is described.

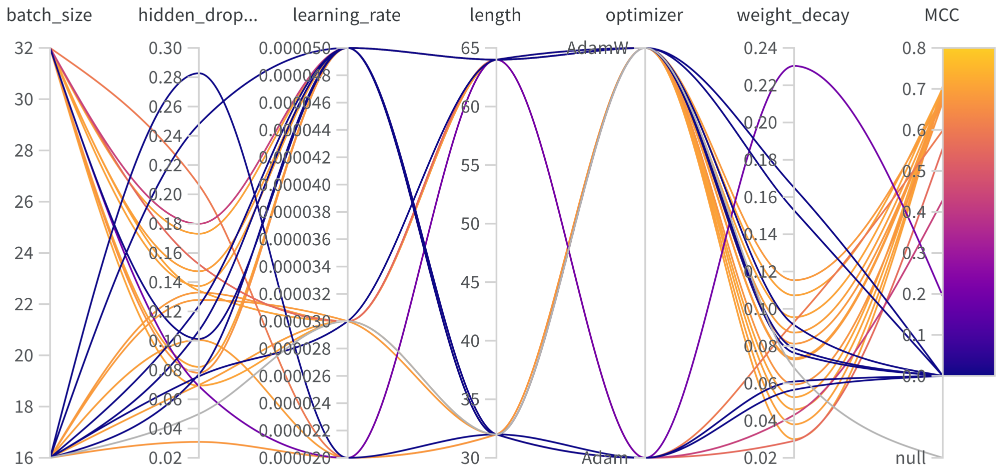


Figure 3.5: Bayesian search visualised for the `MegaAcceptability-BERT-large-uncased` model

Evaluation Metric: Precison, Recall, F1 scores, AUC and PR-curves

Standard evaluation metrics are used for the OpenIE tasks. The metrics are defined below.

Precision: ratio of true positive predictions to the total number of positive predictions made by the model.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: ratio of true positive predictions to the total number of actual positive instances.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 score: harmonic mean of precision and recall, providing a balance between the two for evaluating a model.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

(Precision-Recall) PR-curves: plots the trade-off between precision and recall across different threshold settings.

(Area under the curve) AUC: refers to the area under the PR curve that measures the overall ability of the model to discriminate between positive and negative classes.

Based on the requirements of each of the models, once the acceptability model is ready, the generated triples for each of the benchmarks are joined together and then passed to the fine-tuned/base model.

For the base model, the probability of the joined triple is calculated (depending on the type of model, probability calculations will differ), and then acceptability measures are applied to provide approximate acceptability judgments. For the fine-tuned models, the softmax'ed logits, which provide the probability of the triple being linguistically acceptable, are used and modified by the acceptability measures. Finally, the F1 scores and AUC are generated for each (model, acceptability-measure, benchmark) combination and compared in the next chapter.

As is the case with OpenIE research, P-R curves are generated by analysing the performance of the model across various confidence thresholds. The F1 score presented for each system is determined by an optimal confidence threshold derived from the development set.

The next chapter details the results and analysis of the experiments performed in this chapter.

Chapter 4

Results and Analysis

This chapter is dedicated to the reporting of results from all the experiments as well as an analysis of the results. This includes ablation studies, error analysis and comparison against current State-of-the-art (SoTA) and other established systems in the OpenIE space.

	OIE2016	WEB	NYT	PENN	CaRB
	F1/AUC	F1/AUC	F1/AUC	F1/AUC	F1/AUC
ClausIE	58.8/37.6	44.9/40.1	29.6/22.9	34.6/28.4	45.0/22.4
OpenIE4	59.6/41.7	55.7/40.5	38.3/24.0	42.6/28.1	48.8/27.2
PropS	55.6/33.8	58.9/48.0	37.2/22.1	39.1/27.7	31.9/12.6
DeepEx	72.6/58.6	91.2/82.4	85.50/72.5	88.5/81.5	22.3/4.60
C-N-s	73.11/55.48	92.29/87.04	93.28/88.39	97.07/94.70	24.20/9.20
M-N-s	73.10/54.00	92.74/87.25	92.61/88.49	95.14/90.40	24.5/9.20

Table 4.1: F1 and AUC scores across systems. `C-N-s` stands for `CoLA-NormLP-sub` model. The same pattern is followed for `M-NormLP-sub.`

Experimental Setup

Before presenting the results, a short note on the experimental setup is necessary.

Each benchmark was collected from its source and the models were collected from the HuggingFace Hub [Wolf et al., 2019]. The triples were collected using the `bert-large-cased` model, batch size was set to 4 and the beam search was over 6 tokens. The linguistic acceptability model was trained on a `NVIDIA A100-SXM4-40GB` GPU. The generated triples were joined and then passed to the linguistic acceptability model where they were ranked on the basis of the logits obtained for acceptability. If the model in use was a basic `bert-large-*` model, the probability of the triple according to the language model was calculated. The probability was normalized using the acceptability measures. Finally, the results for each model was reported.

4.1 Results

Table 4.1 presents the best overall results of the proposed system and compares against other popular systems in the OpenIE space. The proposed system surpasses DeepEx by a `0.5` margin for the OIE2016 benchmark, despite using a much smaller amount of data compared to DeepEx for ranking the triples; the proposed system uses 8.5k sentences while DeepEx uses 6.2M sentences. It performs exceptionally well on other evaluation benchmarks as well. However, it does fall behind on the CaRB benchmark, which shall be discussed in section 4.3.

Is there a relation between the length of a sentence and its probability?

Section 3.2 discusses that the length of a sentence may influence its probability, however, the acceptability of a sentence is not influenced by this factor. Looking at the results in Section 4.1.1, it appears that this is the case almost across all benchmarks, hence, normalizing the probability by length seems like a step in the right direction.

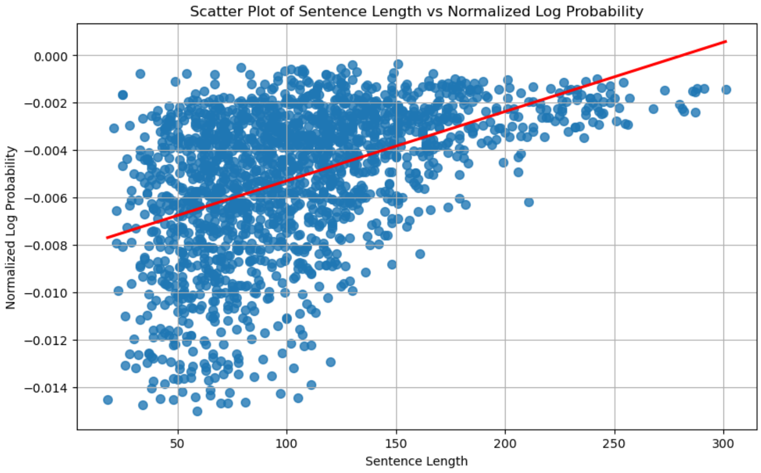


Figure 4.1: `|s|` versus MeanLP, after removing outliers

To further corroborate this result, the relationship between the `MeanLP` and the sentence length was investigated, for the OIE2016 test set (model used is `MegaAcceptability-bert-large-uncased`), revealing a moderately positive correlation. To better visualise the relationship, outliers were removed, the result being in Figure 4.1. Outlier detection was first performed using the Inter-Quartile Range (IQR) method[1].

Spearman's rank correlation coefficient [Zar, 2005] was utilized instead of Pearson's correlation [Cohen et al., 2009] due to its ability in handling non-parametric data. Spearman's correlation is particularly suited for this task because it does not assume a linear relationship between variables, nor does it

[1] https://www.geeksforgeeks.org/interquartile-range-to-detect-outliers-in-data/

require the data to follow a normal distribution. It measures the strength and direction of a monotonic relationship, which is more flexible and further, applicable in this case.

The Spearman’s correlation coefficient is `0.55461856`, with a p-value of 4.969×10^{-15}. This indicates a *moderately positive monotonic* relationship between the variables. The extremely small p-value confirms that this correlation is highly significant and unlikely to have occurred by chance. Consequently, the results validate a meaningful monotonic association between the variables, supporting the use of Spearman’s correlation in this context.

Further, the relationship between sentence length and probability revealed a weak negative correlation (Spearman’s $=$ -0.1188, `p < 0.001`). This indicates that as sentence length increases, there is a slight tendency for the probability assigned by the model to decrease. The statistical significance of the correlation coefficient suggests that this finding is unlikely to be due to random chance.

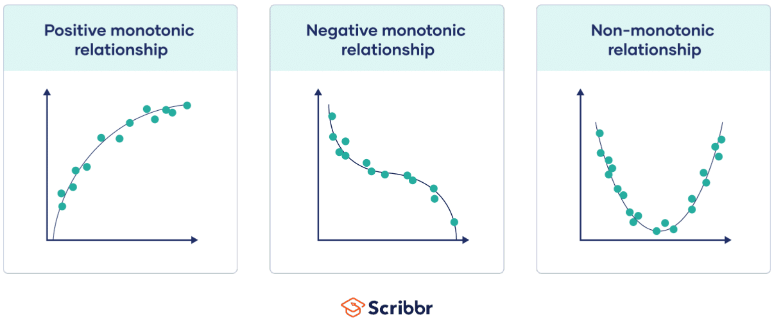


Figure 4.2: Visualization of a positive, negative and absence of a monotonic relationship. Source in footnote.

4.1.1 Results across all tasks

Each of the tables in section 4.1.1 detail the results of experiments conducted for the five benchmarks, across the five normalization methods, with different flavours of the `bert-large` model. The first two tables reflect different variants of the base `bert-large` models, to demonstrate the differences in the results for `cased` versus `uncased` tokenization.

Tables 4.2 and 4.3 show that *augmenting probability scores with acceptability measures are enough to reach competitive scores for the OIE2016 task, even for the basic* `bert-large` *models.* Scores for other benchmark, except for CaRB, are promising as well.

Simply normalizing log probability with the length of the sentence improves the score significantly almost doubling the value, as demonstrated by the score difference between `LogProb` and `MeanLP` for the OIE2016 task in table 4.2. This is true across evaluation benchmarks and models. The score may not double, however, there is a clear jump, even with CaRB. Hence, normalizing the probability with sentence length definitely seems to improve the scores.

`NormLP-sub` is the star of the show, when using this normalization method, the score on the OpenIE task jumps up significantly across most benchmarks

[1] https://www.scribbr.com/statistics/correlation-coefficient/#spearmans-rho

	OIE2016	CaRB	NYT	PENN	WEB
	F1/AUC	F1/AUC	F1/AUC	F1/AUC	F1/AUC
LogProb	33.27/11.44	16.50/3.00	21.47/5.38	46.60/16.81	48.75/22.77
MeanLP	64.68/36.33	26.8/8.2	64.4/36.95	69.90/41.46	77.09/53.90
NLP (sub)	72.21/49.38	25.50/9.60	95.30/92.86	95.04/92.05	92.74/87.69
NLP (div)	63.04/34.27	26.2/7.50	63.08/33.30	69.90/40.255	75.96/52.64
SLOR	50.14/21.38	21.20/4.80	48.99/19.60	71.844/40.13	64.85/38.44

Table 4.2: Results of ranking triples for the downstream OpenIE task on their acceptability by transforming the probabilities obtained from the `bert-large-cased` model into acceptability judgements

	OIE2016	CaRB	NYT	PENN	WEB
	F1/AUC	F1/AUC	F1/AUC	F1/AUC	F1/AUC
LogProb	34.91/13.44	16.6/2.9	21.73/5.63	48.93/22.15	49.65/26.04
MeanLP	65.98/38.91	26.9/8.2	65.10/36.44	67.96/42.33	80.27/61.04
NLP (sub)	72.55/46.46	25.4/9.2	95.30/92.82	95.14/92.45	91.15/84.58
NLP (div)	64.51/36.42	26.0/7.4	63.75/33.59	67.96/42.51	79.59/60.52
SLOR	53.65/25.51	21.9/5.1	51.00/21.80	60.19/31.02	69.16/45.53

Table 4.3: Results of ranking triples for the downstream OpenIE task on their acceptability by transforming the probabilities obtained from the `bert-large-uncased` model into acceptability judgments

and models. Here, too, CaRB is an exception. `NormLP-sub` normalizes the log probability of the sentence against the log unigram probability of the sentence. This measure attempts to discount the affect that more common words have on the probability of sentence. Basically, it tries to bring the sentences to a common ground by not allowing the influence of more/less popular words in the vocabulary. As Lau et al. [2017] comment *...it is a key element in any model that attempts to account for the confounding effect of lexical frequency on acceptability.*

The unigram probability of the sentence does not consider the word order or

	OIE2016	CaRB	NYT	PENN	WEB
	F1/AUC	F1/AUC	F1/AUC	F1/AUC	F1/AUC
LogProb	46.43/22.35	24.5/7.8	40.40/17.13	51.54/26.98	60.15/34.79
MeanLP	60.32/34.63	28.9/10.9	63.75/41.75	79.611/63.1	78.09/59.57
NLP (sub)	73.06/53.98	24.3/9.5	92.92/88.1	95.14/93.19	92.74/8701
NLP (div)	59.79/34.17	28.6/10.8	59.73/35.5	81.55/70.53	76.73/57.29
SLOR	24.84/5.80	8.50/1.20	21.09/5.32	39.58/20.29	44.92/21.99

Table 4.4: Results of ranking triples for the downstream OpenIE task on their acceptability by transforming the probabilities obtained from the CoLA fine-tuned `bert-large-uncased` model into acceptability judgements

	OIE2016	CaRB	NYT	PENN	WEB
	F1/AUC	F1/AUC	F1/AUC	F1/AUC	F1/AUC
LogProb	47.60/24.26	22.3/6.3	42.42/21.44	60.19/42.30	66.66/46.06
MeanLP	64.68/41.39	27.7/1.0	77.18/57.38	83.49/69.49	81.85/67.81
NLP (sub)	73.11/55.48	24.2/9.2	93.28/88.39	97.07/94.70	92.29/87.04
NLP (div)	63.80/40.10	27.1/9.6	78.52/56.75	87.37/74.95	81.40/67.34
SLOR	24.16/6.33	8.40/0.8	21.67/5.03	37.89/15.09	43.63/22.68

Table 4.5: Results of ranking triples for the downstream OpenIE task on their acceptability by transforming the probabilities obtained from the CoLA fine-tuned `bert-large-cased` model into acceptability judgements

context, it simply multiplies the probabilities of the individual words together. By contrast, the probability according to the language model adds in extra information such as context, word dependencies and syntactic and semantic relationships in the sentence. So, a higher score on this metric would mean that the base/fine-tuned model has a more complex/richer understanding of the dependencies between the words in the sentence and further points towards a more acceptable sentence by virtue of the data used to train the model.

CaRB benchmark proves to be an outlier in these experiments since it is no-

	OIE2016	CaRB	NYT	PENN	WEB
	F1/AUC	F1/AUC	F1/AUC	F1/AUC	F1/AUC
LogProb	55.69/34.15	27.7/9.7	57.85/34.10	58.25/25.25	63.11/37.88
MeanLP	69.34/49.25	29.5/11.4	85.23/68.92	93.20/80.21	85.94/70.50
NLP (sub)	73.03/54.83	24.6/9.4	93.28/89.30	97.08/94.44	92.29/87.23
NLP (div)	69.19/48.76	29.3/11.1	85.90/70.84	93.20/79.93	86.39/71.47
SLOR	24.78/6.82	8.7/1.2	22.14/5.59	38.29/18.03	45.30/24.53

Table 4.6: Results of ranking triples for the downstream OpenIE task on their acceptability by transforming the probabilities obtained from the MegaAcceptability fine-tuned `bert-large-cased` model into acceptability judgements

	OIE2016	CaRB	NYT	PENN	WEB
	F1/AUC	F1/AUC	F1/AUC	F1/AUC	F1/AUC
LogProb	56.38/35.31	29.4/10.9	50.67/23.75	60.78/28.83	65.38/42.11
MeanLP	69.69/49.32	29.7/12.1	82.55/66.96	89.32/69.16	86.16/69.35
NLP (sub)	73.10/54.00	24.5/9.2	92.61/88.49	95.14/90.40	92.74/87.25
NLP (div)	69.73/49.53	29.7/12.2	83.22/67.33	89.32/68.99	87.07/71.34
SLOR	25.54/6.73	8.8/1.6	22.31/5.80	38.29/18.05	46.40/22.81

Table 4.7: Results of ranking triples for the downstream OpenIE task on their acceptability by transforming the probabilities obtained from the MegaAcceptability fine-tuned `bert-large-uncased` model into acceptability judgments

ticeable that `NormLP-div` or `MeanLP` has higher scores compared to `NormLP-sub`. The score might be marginally better, but there is always a clear improvement.

`SLOR` as an acceptability measure should have ideally been the best since it uses both the length of the sentence and the normalized log unigram probability. Further, `SLOR` is widely regarded as a **fluency** metric. Surprisingly, `SLOR` performs better for the base models, with the F1 score nearly halving for fine-tuned models. For the CaRB benchmark, the scores on the `SLOR` metric are abysmal for all models except the based models.

Considering the formula used for SLOR: since the normalized log unigram probability and the length of the sentence remains the same, this indicates that the base model does a better job at ranking sentences higher (by assigning higher probabilities) of those examples which would perform better on the OpenIE task.

A curious observation is the similarity between the scores of `MeanLP` and

NormLP_div across many benchmarks and models. MeanLP is the probability of a model normalized by its length while NormLP_div is the probability of a model normalized by the unigram probability of the sentence according to the model. The reason for this similarity is unclear.

In general, the trend seems to be that fine-tuning on CoLA or the MegaAcceptability dataset does seem to have positive/neutral impact across most normalization measures (except for SLOR) and datasets. For example, looking at the scores produced by the PENN dataset, the F1 score consistently increases as more linguistically acceptable data is provided to the model. This largely holds true for the OIE2016 task as well except for two notable exceptions: SLOR and NormLP_sub. For the former, the F1 score gets halved, while for the later, despite more data, better fine-tuning, the score hits a ceiling of 73.xx. This is the case even after conducting an ablation study that removes most layers of the fine-tuned model, described in Section 4.2

The CaRB evaluation benchmark is a special case, it does not use the same evaluation script as the other four benchmarks, but it uses the same dataset(with different annotations) as the OIE2016 dataset. The low scores of the DeepEx model and of the proposed system on the CaRB benchmark need further investigation, which is discussed in section 4.3.

What could be the reason for NYT and PENN performing so well?

The results of NYT and PENN are very impressive, surpassing DeepEx, the current SoTA (according to URL[2]). The initial hypothesis was that, for some sentences, the entire sentence was being predicted as a triple. As in, the subject and object noun phrases that were ranked the highest were periphery noun phrases, basically, noun phrases at the beginning and the end of the sentence. The relation that was picked for this pair of noun phrases was the entirety of the sentence between them.

Acceptability Measure	(matched triples, total sentences)
SLOR	(0.0, 52.0)
MeanLP	(11.0, 52.0)
LogProb	(3.0, 52.0)
NormLP_div	(11.0, 52.0)
NormLP_sub	(8.0, 52.0)

Table 4.8: Fuzzy matching of joined triples against sentences for the CoLA-bert-large-cased model on the PENN evaluation benchmark. The results for this benchmark-model combination are presented in Table 4.5

A small experiment was performed to check this hypothesis. The CoLA-bert-large-cased model was chosen, since the performance is one of the highest for this variant (F1: 97.07/ AUC: 94.70). The evaluation benchmark chosen is PENN. Fuzzy matching using the fuzzywuzzy [SeatGeek, 2024] library was used to compare the joined triple with the original sentence. The fuzz-ratio threshold was set to 95. This threshold was chosen on the basis that, a higher

[2] https://paperswithcode.com/task/open-information-extraction

threshold may not capture all triples that are very similar to their sentences, while a lower threshold might catch triples that are relevant. Keeping a high threshold seems reasonable, after all, the triples are parts of the sentence.

From Table 4.8 it is clear that nearly 15% of triples for the best performing normalization method (NormLP_sub) are very similar to their sentences. Hence, there is a non-significant number of triples that are similar to the sentences they are extracted from.

What is the performance of non-BERT acceptability models (on the OpenIE task) like?

An interesting question would be assessing the performance of non-BERT models. Lau et al. [2020] have provided unigram probabilities for the GPT2-medium [Radford et al., 2019] model. GPT2-medium is not bi-directional like BERT and so, the probability calculations were pretty straightforward. The results for the OIE2016 benchmark, across all acceptability measures is presented below.

Acceptability Measures	**F1**	**AUC**
LogProb	17.03	2.15
SLOR	46.63	17.98
NormLP_div	61.51	39.20
MeanLP	64.08	44.16
NormLP_sub	72.90	61.25

Table 4.9: Performance of GPT2-medium model on the OIE2016 benchmark for each acceptability measure.

Even for `GPT2-medium`, which is a generative model and not a masked language model, the results for the `NormLP_sub` acceptability measure are the highest, again surpassing the result obtained by DeepEx. These results provide evidence that the proposed method can be replicated for other LLMs while maintaing performance.

The Wang et al. [2021] paper on DeepEx reports scores for only the OIE2016, NYT, PENN and WEB benchmarks. The score on the CaRB dataset was delegated to future work. This has been completed and a PR has been opened in the DeepEx codebase. The pull-request can be accessed here[3].

The results for the OIE2016 and CaRB benchmarks are the scores when the top-3 triples are chosen, while for the rest of the benchmarks, it is the top-1 triple that is chosen. The choice for k in top-k is partly due the DeepEx's preference for the same and partly due to the best results presenting for these benchmarks on using $k = 1/k = 3$.

4.2 Studies

Based on the results tabulated in section 4, it can be inferred that despite the simplicity of the system, which consists of a linguistic acceptability dataset that

[3] `https://github.com/wang-research-lab/deepex/pull/20`

has just 8500 data points, it is possible to achieve performance that is competitive with, and in some cases, surpasses the results obtained by the DeepEx contrastive ranking model. It would be interesting to conduct an ablation study (Section 4.2.1) to identify the components of this system that impact the performance or that can be pruned to either achieve a comparable score or a better score. This study is further essential, since Table 4.2 also shows that simple acceptability measures applied to probability values generated by the base `bert-large` model were also competitive.

In general, ablation studies are also essential to the development of systems since they provide an insight into the reliability, robustness and necessity of various components. In the case of the proposed system, the component with the most variability (in terms of performance) is the fine-tuned linguistic acceptability model.

Combinations of different kinds of datasets (in terms of size, domain, linguistic variability) and different models (pre-training objectives, number of parameters, pre-training data) would perform in their own unique manner, which is further influenced by the fine-tuning methods used. The work in this thesis has been intentionally kept simple, in terms of the model and the fine-tuning procedures, to showcase the performance of a minimal system. Therefore, dissecting the fine-tuned model would be an insightful form of ablation study, in part, to see exactly what component of the system contributes to the results, but also, to see if the system could be further simplified.

The next study (Section 4.2.2) in this section involves the connection between SRL (Semantic Role Labelling) and OpenIE. SRL performs a very similar task to OpenIE, so much so that, an algorithm was used to convert a QA-SRL dataset to an OpenIE one (OIE2016). Both methods extract two entities and the relation between them. A significant difference between the two would be the fact that SRL explicitly labels the semantic relationships for each entity extracted.

To add a semantic aspect to this thesis, a small study is conducted, which showcases an alternative method of extracting OpenIE triples using the results of a pre-trained SRL model.

4.2.1 Study 1: Freezing layers of the linguistic acceptability model

This is an ablation study. Lee et al. [2019] in their paper *What Would Elsa Do? Freezing Layers During Transformer Fine-Tuning* perform experiments to demonstrate the effect of freezing different layers of the pre-trained model on various downstream tasks, one of them being the linguistic acceptability task (CoLA dataset). They freeze the embeddings and weights upto the 0th, 18th and 24th layer and compare the performance of each of these systems to the full-version where no layers are frozen. Their results mention a substantial drop in the MCC score across the experiments for the CoLA dataset while other tasks seem to retain a significant amount of performance, sometimes even upto 90% of the score of the fully fine-tuned model. This brings up a relevant question for our experiment:

> *How would freezing different layers in the linguistic acceptability model affect the performance of the OpenIE task?*

To answer this, the `MegaAcceptability-bert-large-cased` model is used as a guinea pig. The reason this model is chosen among the 4 that have been fine-tuned in this thesis is due to the fact that this model has the best overall performance across all normalization methods for the OIE2016 dataset. Moreover, performing these experiments for all the four models would prove computationally infeasible. Also, since the CoLA dataset has been examined in Lee et al. [2019], it would be interesting to see if the MegaAcceptability dataset follows a similar trend (regarding the drop in performance).

Five experiments are conducted: for each model, 0, 6, 12, 18, 24 layers are frozen, in addition to the embedding layer. The fine-tuning hyper-parameters were kept the same, following the same methodology as Lee et al. [2019].

The results for the `NormLP_sub` method is reported, since this method generally has the highest F1 score across datasets and models. The F1 and AUC for the OIE2016 evaluation dataset is reported in table 4.10.

Frozen upto layer-	MCC	F1/AUC
no freezing	68.51	73.03/54.83
0th	68.46	73.06/54.74
6th	69.01	73.03/55.10
12th	67.73	73.14/55.20
18th	64.18	73.08/55.35
24th	47.71	73.12/54.69

Table 4.10: MCC score after freezing layers of the `MegaAcceptability-bert-large-cased` model, the parameters that were activated for each model and the corresponding F1 score on the OIE2016 OpenIE task. The embedding layer is frozen for all experiments.

Results

As table 4.10 shows, there isn't substantial change in the MCC score when 0, 6, 12 layers are frozen, the score remains relatively stable. A notable decrease in the MCC score is observed only when 18 layers are frozen. Freezing all the layers does affect the MCC, the drop of 20 points is significant.

What could be the reason that most layers are not necessary to match the results (on the linguistic acceptability task) as the entire model for the MegaAcceptability dataset? One guess could be the fact that the MegaAcceptability dataset is far more **homogeneous** in nature, with no specific domain defined, while the OOD (out-of-domain) dev set in the CoLA dataset is quite diverse. The homogeneity of the MegaAcceptability dataset is due to the fact that the focus of each sentence in the dataset is around its verb and the lexical content is minimal. This is further described in Section 3.1.1. The diversity of the OOD CoLA dev set is mentioned in Warstadt et al. [2019].

Hence, for the CoLA dataset,more layers are required to truly understand the task. However, we can also discount the difference in the MCC's, the drop from the 0th-18th layer in the CoLA dataset is of around 10 points, while for the same interval, the drop in MCC for the MegaAcceptability dataset is about 4 points.

The difference is still significant, but could possibly be explained by the similarity of domain in the MegaAcceptability dataset.

The F1/AUC barely changes even after freezing all layers. This result is somewhat expected since the results in tables 4.3 and 4.2, where no task-specific fine-tuning occurs are comparable with the results after fine-tuning on the CoLA and MegaAcceptability dataset. These results further strengthen the analysis that *fine-tuning on more data doesn't necessarily co-relate to a much higher OpenIE task score, specifically for the `NormLP_sub` measure* .

As mentioned in the previous sections, it seems that despite more data, the F1 score reaches a ceiling beyond which it doesn't seem to increase. Of course, this could be possibly rectified by better/more complex models/better fine-tuning strategies than the one employed, but for the parameters of the experiment within this thesis, the F1 score does not seem to increase beyond `73.xx`.

4.2.2 Study 2: Semantic Role Labelling for OpenIE

Semantic Role Labelling (SRL) picks the *predicate-argument* structure and fixes "labels" to them that identifies if they are the doer, the action or the receiver in a given sentence. Considering this succinct definition, and the relation to the OIE2016 dataset (the dataset is created using another QA-SRL dataset), it would be interesting to see an application of SRL for OpenIE. OpenIE is supposed to extract relevant triples and sort them, while SRL labels the subject, verbs and object.

The implementation of this study is quite simple: use a pre-trained SRL-BERT (base) [Shi and Lin, 2019] model supplied by AllenNLP [Gardner et al., 2018] to identify verbs and their surrounding arguments within sentences, then collect context information for each identified verb and finally extract specific argument types and construct triples comprising the left context, verb, and right context.

The results are reported for all the 5 evaluation datasets. All extracted triples are chosen for CaRB/OIE2016 while one triple is chosen for the rest of the evaluation datasets. Unfortunately, there is a gap in this method: that of the ranker. The AllenNLP model does not provide any metric by which to judge one extraction as more efficient than another. So, for now, all triples are chosen in random order. The number of triples extracted generally number around 3 or `< 3`, so this is not a significant issue yet.

The results for this task are displayed in table 4.2.2. The fact that OIE2016 performs well is trivial, the dataset was built by applying an algorithm over a QA-SRL dataset, it is expected that a model that extracts semantic labels would do a good job on this dataset+scorer combination.

The CaRB benchmark uses the same dataset, however, the annotations are performed by humans and the scorer is also different. Here, the score drops as compared to OIE2016, but the system still performs better than DeepEx and the proposed system's performance on CaRB.

For the rest of the benchmarks, the results are competitive with most systems in the OpenIE space.

Evaluation benchmark	F1	AUC
OIE2016	73.86	61.35
CaRB	35.2	28.8
NYT	66.01	54.48
PENN	72.78	62.16
WEB	84.04	75.20

Table 4.11: Results for the study in section 4.2.2. This table displays the results of adapting the results of the BERT-SRL model for the OpenIE task.

4.3 Error Analysis

The errors in the DeepEx model would propagate to any derivative of its system, in this case, the errors from the generation step would be present in the proposed system as well. The authors of Wang et al. [2021] do not mention errors concerning the ranking step, most of the error analysis in the paper was devoted to the generation step.

Specifically for OpenIE, DeepEx identifies that the majority of the errors in this step were due to the incorrect assignment of arguments by the Spacy noun-chunker, while 10% of the errors are due to *long sentences*. The errors mentioned due to long sentences are taken to mean that the input sentence was long, while the triple extracted only focused on a single small part of the sentence. This can be inferred from the example presented in the appendix of the paper.

However, DeepEx does not perform as well as expected on the CaRB benchmark, despite stellar performance on the other benchmarks, including on OIE2016.

What could be the possible reason for this result?

One of the differences between the two evaluation benchmarks is the scorer. The authors of CaRB have put forward the drawbacks they notice in the OIE2016 scorer, some of them are listed below. It would be interesting to draw a contrast between the proposed system and DeepEx on the CaRB benchmark, since this is the benchmark on which both of the systems fail to achieve competitive scores.

Table 4.3 details a partial list of the errors that could have been the cause of the drop in results on the CaRB benchmark as compared to OIE2016. The second column is paraphrased from the CaRB paper. Additional notes on the errors follow:

- For the first error, another reason as to why the DeepEx system falls behind is that, there is a possibility of ”nesting” of NP pairs, that is, one NP pair is contained within another. If the gap between the NP pairs is not too large, this could lead to redundant extractions.

- For the second error, lexical matching also favours the proposed system greatly. A system that optimizes for linguistic acceptability would favour sentences or clauses that would be complete units. This tends to result in longer sentences. Longer sentences would translate to better lexcial matching, since more words would be matched to a particular tuple on average. This error is circumvented by the CaRB benchmark using exact tuple matching.

Index	Difference b/w OIE2016 and CaRB scorer	Is DeepEx affected?	Is the proposed system affected?
1	*"...By single matching for precision, CaRB penalizes Open IE systems that produce several very similar and redundant extractions."*	Yes, since the beam size is set to 6, for a given triple, 6 samples are possibly passed to the ranker. If the ranker chooses more than one from the 6 samples, it is possible to have more than one top-3 triples with the same `ARG1` and `ARG2`.	Yes, since the proposed system simply joins the triples and does not concern itself with the `ARG1` and `ARG2`. It simply sorts by acceptability.
2	*"...Another significant change from OIE2016 scorer is in the use of tuple match instead of lexical match."*	Yes, since the noun-chunker is a fault line that has been already identified, it is possible that lexically matching the whole triple as a sentence might be more lenient than strict tuple matching.	This error propagates here as well.
3	*"...This scorer has been identified to not penalize long extractions."*	Yes. The relations are chosen based on highest attention scores between two arguments. This encourages the addition of more words to a triple, since more the number of words, higher the attention score. So for a given argument pair, it is likely that all the words between the two of them may be chosen as the relation, since choosing more words might potentially lead to higher score.	Yes. An acceptability model will sort well-formed clauses and sentences higher. It is not unlikely for a longer sentence to be favoured more here.

4.4 Discussion

As a final note, the proposed system performs fairly well across diverse domains and benchmarks, with lesser amount of data points. Further, the experiments also suggest that converting the probability scores to acceptability judgements do produce better results for the OpenIE task, even on out of the box models. It might be possible to draw the conclusion that acceptability has an effect on OpenIE triples, and this factor can be used as one of the extraction methods for future OpenIE systems.

This work also brings to light the drawbacks of various evaluation benchmarks, particularly that of OIE2016. Comparing OIE2016 and the CaRB benchmark is a particularly interesting case due to the similarity in the test sentences and the dissimilarity in the scorer as well as the annotation methodology. Human annotation is considered superior to the one derived through an algorithm, as is the case with OIE2016. Further, the authors of CaRB, Bhardwaj et al. [2019] meticulously critique the leniency of the OIE2016 scorer. Considering these two points, comparing the scores between OIE2016 and CaRB provide precious insight into the quality of the extractions.

DeepEx and the proposed system perform well on the scorer implemented by Stanovsky et al. [2018a], but the results on CaRB are not competitive with other models.

Some of the errors that the proposed method has could be attributed to the triples generated by DeepEx. It is possible that triples generated by another method could produce better performance.

The performance on the PENN and NYT benchmarks is admirable, however, around 15% of the joined triples are at least 95% similar to the sentence. Hence, adding a filter that would remove joined triples that are similar to the sentence may subdue the results.

Overall, a faithful study on the relation between the linguistic acceptability of a triple and its impact on the OpenIE F1 score has been conducted. The results have been reported and analysed.

There are some potential points of failure here, one of them could be the methods used to create the acceptability models. There is always a chance of better, bigger architectures with better data quality producing more linguistically acceptable sentences.

4.5 Limitations and Future Work

The primary limitation of this method is that it can only be applied to languages that follow the SVO word order. Modifications need to be made to the application to other languages. The thesis is also limited by the quality of the generated triples. If the generated triples have flaws, this trickles down to affect the ranker, too. Finally, the availability of unigram probabilities also restricts the scope of this thesis. The creation of unigram probabilities for other LLMs is earmarked for future work.

Conclusion

OpenIE is a relevant and challenging task. There is an ever growing need for better tuple extraction from large amounts of text in an unsupervised manner. The applications using such tuples range across fields and domains, and as such, better models, methods and systems are always in high demand.

The method proposed in this thesis shows promising results across benchmarks and datasets, despite using a very small amount of training data. The results obtained using the proposed method have been thoroughly analysed on various models as well, providing evidence that the method can be used in a model agnostic manner.

The drawbacks concerning the method have been demonstrated using error analysis and ablation studies.

Bibliography

Hannah Youngeun An and Aaron Steven White. The lexical and grammatical sources of neg-raising inferences. *arXiv preprint arXiv:1908.05253*, 2019.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 344–354, 2015.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Michele Banko and Oren Etzioni. The tradeoffs between open and traditional relation extraction. In *Proceedings of ACL-08: HLT*, pages 28–36, 2008.

Sangnie Bhardwaj, Samarth Aggarwal, et al. Carb: A crowdsourced benchmark for open ie. 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)*, pages 6262–6267, 2019.

John Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. *Advances in neural information processing systems*, 2, 1989.

Noam Chomsky. *Aspects of the Theory of Syntax*. Number 11. MIT press, 1965.

Noam Chomsky. The logical structure of linguistic theory. 1975.

Janara Christensen, Stephen Soderland, Oren Etzioni, et al. Semantic role labeling for open information extraction. In *Proceedings of the NAACL HLT 2010 first international workshop on formalisms and methodology for learning by reading*, pages 52–60, 2010.

Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise reduction in speech processing*, pages 1–4, 2009.

Common Crawl. Common crawl. `http://commoncrawl.org`. Accessed: 2024-07-17.

Luciano Del Corro and Rainer Gemulla. Clausie: clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366, 2013.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Pamela Downing and Michael Noonan. *Word order in discourse*, volume 30. John Benjamins Publishing Company, 1995.

Matthew S Dryer. Word order. *Language typology and syntactic description*, 1: 61–131, 2007.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Steven Y. Feng, Varun Gangal, Dongyeop Kang, Teruko Mitamura, and Eduard Hovy. GenAug: Data augmentation for finetuning text generators. In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić, editors, *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 29–42, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.deelio-1.4. URL `https://aclanthology.org/2020.deelio-1.4`.

A Field. Discovering statistics using spss. 3rd edlos angeles, 2009.

John Fields, Kevin Chovanec, and Praveen Madiraju. A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe? *IEEE Access*, 2024.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. Allennlp: A deep semantic natural language processing platform. *arXiv preprint arXiv:1803.07640*, 2018.

Siddhant Garg and Goutham Ramakrishnan. Bae: Bert-based adversarial examples for text classification. *arXiv preprint arXiv:2004.01970*, 2020.

Santiago González-Carvajal and Eduardo C Garrido-Merchán. Comparing bert against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*, 2020.

Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 643–653, 2015.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *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)*, pages 4129–4138. Association for Computational Linguistics, June 2019. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419.

Matthew Honnibal, Ines Montani, and spaCy contributors. spacy: Industrial-strength natural language processing in python, 2015–. URL https://spacy.io.

Yuening Jia. *Journal of Physics: Conference Series*, 1314(1):012186, 2019. doi: 10.1088/1742-6596/1314/1/012186. URL https://commons.wikimedia.org/w/index.php?curid=121340680. CC BY-SA 3.0.

Hanlei Jin, Yang Zhang, Dan Meng, Jun Wang, and Jinghua Tan. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv preprint arXiv:2403.02901*, 2024.

Katharina Kann, Sascha Rothe, and Katja Filippova. Sentence-level fluency evaluation: References help, but can be spared! *arXiv preprint arXiv:1809.08731*, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ronald W Langacker. How to build an english clause. *Journal of Foreign Language Teaching and Applied Linguistics*, 2(2):1–45, 2015.

LanguageTooler. language-check: A python wrapper for languagetool. https://github.com/myint/language-check, n.d. Accessed: 06.07.2024.

Jey Han Lau, Alexander Clark, and Shalom Lappin. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive science*, 41(5):1202–1241, 2017.

Jey Han Lau, Carlos Armendariz, Shalom Lappin, Matthew Purver, and Chang Shu. How furiously can colorless green ideas sleep? sentence acceptability in context. *Transactions of the Association for Computational Linguistics*, 8: 296–310, 2020.

Jaejun Lee, Raphael Tang, and Jimmy Lin. What would elsa do? freezing layers during transformer fine-tuning. *arXiv preprint arXiv:1911.03090*, 2019.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

Pai Liu, Wenyang Gao, Wenjie Dong, Songfang Huang, and Yue Zhang. Open information extraction from 2007 to 2022 – a survey, 2022. URL https://arxiv.org/abs/2208.08690.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19 (2):313–330, 1993.

B.W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405(2):442–451, October 1975. ISSN 0005-2795. doi: 10.1016/0005-2795(75) 90109-9. URL `http://dx.doi.org/10.1016/0005-2795(75)90109-9`.

Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, 2013.

Iqra Muhammad, Anna Kearney, Carrol Gamble, Frans Coenen, and Paula Williamson. Open information extraction for knowledge graph construction. In *Database and Expert Systems Applications: DEXA 2020 International Workshops BIOKDD, IWCFS and MLKgraphs, Bratislava, Slovakia, September 14–17, 2020, Proceedings 31*, pages 103–113. Springer, 2020.

Adam Pauls and Dan Klein. Large-scale syntactic language modeling with treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 959–968, 2012.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611*, 2020.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III 21*, pages 148–163. Springer, 2010.

Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J Liu, Sharan Narang, Wei Li, and Yanqi Zhou. Exploring the limits of transfer learning with a unified text-to-text transformer. *Google, Tech. Rep.*, 2019.

Evan Sandhaus. The New York Times Annotated Corpus, 2008. URL `https://hdl.handle.net/11272.1/AB2/GZC6PL`.

SeatGeek. fuzzywuzzy: Fuzzy string matching in python, 2024. URL `https://github.com/seatgeek/fuzzywuzzy`. Accessed: 2024-07-17.

Murray Shanahan. Talking about large language models. *Communications of the ACM*, 67(2):68–79, 2024.

Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3-4):591–611, 1965.

Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019.

Gabriel Stanovsky and Ido Dagan. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1252. URL `https://doi.org/10.18653/v1/d16-1252`.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, page (to appear), New Orleans, Louisiana, June 2018a. Association for Computational Linguistics.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 885–895, 2018b.

Murat Tezgider, Beytullah Yildiz, and Galip Aydin. Text classification using improved bidirectional transformer. *Concurrency and Computation: Practice and Experience*, 34(9):e6486, 2022.

Russell S Tomlin. Basic word order: Functional principles: Croom helm london. *NCls–Noun Classifier NNum-Noun Numeral Nom-Nominative Rel-Relative NRel-Noun Relative RelN-Relative Noun*, 1986.

M Onat Topal, Anil Bas, and Imke van Heerden. Exploring transformers in natural language generation: Gpt, bert, and xlnet. *arXiv preprint arXiv:2102.08036*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. Zero-shot information extraction as a unified text-to-triple translation. *arXiv preprint arXiv:2109.11171*, 2021.

Chenguang Wang, Xiao Liu, and Dawn Song. Ielm: An open information extraction benchmark for pre-trained language models, 2022. URL `https://arxiv.org/abs/2210.14128`.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Aaron Steven White and Kyle Rawlins. A computational model of s-selection. In *Semantics and linguistic theory*, pages 641–663, 2016.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. Emerging cross-lingual structure in pretrained language models. *arXiv preprint arXiv:1911.01464*, 2019.

Ying Xu, Mi-Young Kim, Kevin M Quinn, Randy Goebel, and Denilson Barbosa. Open information extraction with tree kernels. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 868–877, 2013.

Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. Bert representations for video question answering. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1556–1565, 2020.

Alexander Yates, Michele Banko, Matthew Broadhead, Michael Cafarella, Oren Etzioni, and Stephen Soderland. TextRunner: Open information extraction on the web. In Bob Carpenter, Amanda Stent, and Jason D. Williams, editors, *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, Rochester, New York, USA, April 2007. Association for Computational Linguistics. URL `https://aclanthology.org/N07-4013`.

Jerrold H Zar. Spearman rank correlation. *Encyclopedia of Biostatistics*, 7, 2005.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

List of Figures

List of Tables

List of Abbreviations

Appendix A

Additional Material

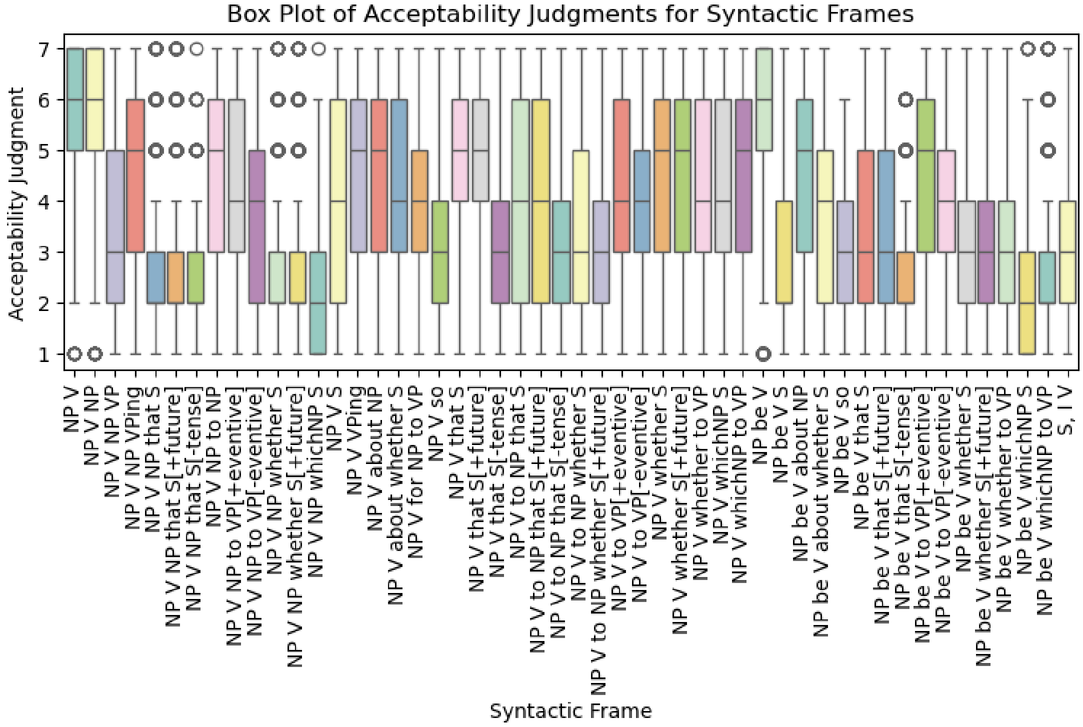
Table A.1: Syntactic Frames, Means, and Counts for the MegaAcceptability dataset

Frame	Mean ± Std	Count
NP V NP	5.67 ± 1.36	2479
NP V	5.5 ± 1.37	2338
NP be V	5.45 ± 1.65	2236
NP V that S	4.89 ± 1.66	1948
NP V that S[+future]	4.83 ± 1.64	1930
NP V whether S[+future]	4.52 ± 1.61	1816
NP V VPing	4.48 ± 1.57	1780
NP V NP VPing	4.41 ± 1.58	1780
NP V about NP	4.72 ± 1.74	1772
NP V NP to NP	4.55 ± 1.71	1768
NP be V about NP	4.62 ± 1.85	1762
NP V whether S	4.54 ± 1.66	1749
NP V whichNP to VP	4.46 ± 1.66	1742
NP be V to VP[+eventive]	4.41 ± 1.74	1702
NP be V to VP[-eventive]	4.19 ± 1.55	1684
NP V whichNP S	4.26 ± 1.66	1678
NP V about whether S	4.3 ± 1.68	1630
NP V whether to VP	4.23 ± 1.62	1626
NP V to VP[+eventive]	4.21 ± 1.69	1586
NP V to NP that S[+future]	4.02 ± 1.81	1552
NP V S	4.0 ± 1.83	1522
NP V for NP to VP	3.96 ± 1.63	1520
NP V NP to VP[+eventive]	4.17 ± 1.78	1516
NP V to NP that S	4.01 ± 1.88	1508
NP be V about whether S	3.83 ± 1.8	1468
NP V to VP[-eventive]	3.87 ± 1.61	1448
NP V to NP whether S[+future]	3.31 ± 1.48	1404
NP be V that S	3.67 ± 1.92	1402
NP V NP to VP[-eventive]	3.62 ± 1.55	1394

Continued on next page

Frame	Mean ± Std	Count
NP V that S[-tense]	3.48 ± 1.31	1382
NP be V that S[+future]	3.54 ± 1.79	1366
NP V to NP whether S	3.51 ± 1.67	1360
NP V NP VP	3.43 ± 1.55	1314
NP V so	3.19 ± 1.29	1272
NP be V S	3.03 ± 1.71	1262
NP be V whether S[+future]	3.13 ± 1.61	1252
NP be V whether to VP	3.2 ± 1.55	1244
NP be V whether S	3.08 ± 1.56	1227
NP V to NP that S[-tense]	2.97 ± 1.39	1212
S, I V	3.16 ± 1.46	1170
NP be V so	2.85 ± 1.16	1162
NP V NP that S	2.76 ± 1.51	1136
NP be V whichNP to VP	2.7 ± 1.39	1134
NP be V that S[-tense]	2.64 ± 1.24	1130
NP V NP that S[+future]	2.77 ± 1.52	1120
NP V NP whether S[+future]	2.59 ± 1.34	1098
NP V NP whether S	2.5 ± 1.32	1094
NP be V whichNP S	2.34 ± 1.24	1072
NP V NP whichNP S	2.12 ± 1.05	1042
NP V NP that S[-tense]	2.23 ± 1.0	1038

Figure A.1: Acceptability Judgements of Syntactic Frames for MegaAcceptability dataset



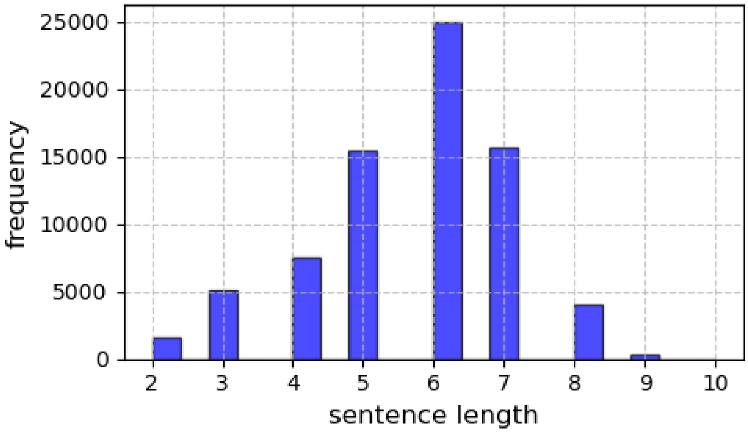


Figure A.2: For the MegaAcceptability dataset: Frequency versus sentence length

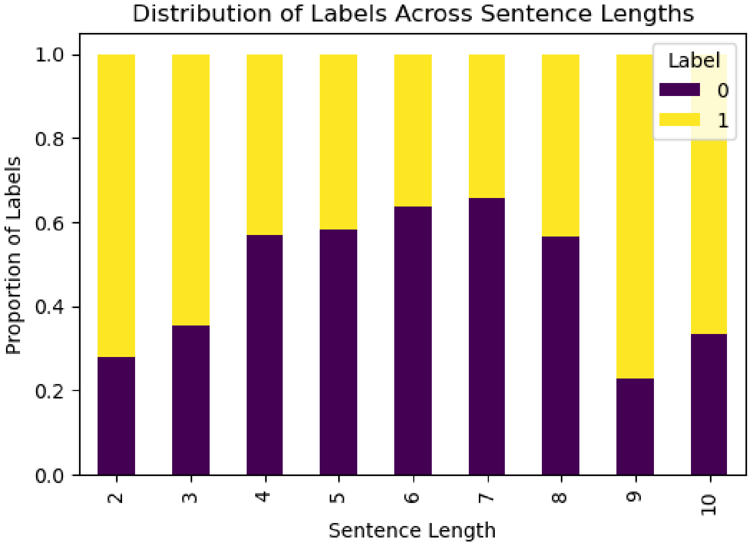


Figure A.3: For the MegaAcceptability dataset: The distribution of unacceptable and acceptable sentences across sentence lengths