MULTI-LANGUAGE INFORMATION EXTRACTION WITH TEXT PATTERN RECOGNITION

Johannes Lindén, Tingting Zhang, Stefan Forsström and Patrik Österberg

Department of Information System and Technology, Mid.
Sweden University, Sundsvall, Sweden

ABSTRACT

Information extraction is a task that can extract meta-data information from text. The research in this article proposes a new information extraction algorithm called GenerateIE. The proposed algorithm identifies pairs of entities and relations described in a piece of text. The extracted meta-data is useful in many areas, but within this research the focus is to use them in news-media contexts to provide the gist of the written articles for analytics and paraphrasing of news information. GenerateIE algorithm is compared with existing state of the art algorithms with two benefits. Firstly, the GenerateIE provides the co-referenced word as the entity instead of using he, she, it, etc. which is more beneficial for knowledge graphs. Secondly GenerateIE can be applied on multiple languages without changing the algorithm itself apart from the underlying natural language text-parsing. Furthermore, the performance of GenerateIE compared with state-of-the-art algorithms is not significantly better, but it offers competitive results.

KEYWORDS

Information Extraction, IE, Information representation, Knowledge Graph, Natural Language Processing, NLP, Pattern Recognition, Entity Recognition.

1. INTRODUCTION

Modelling data with machine learning algorithms has shown promising results in various areas, such as image processing, robotics and natural language processing. These areas are growing, both in size and number, and machine learning becomes more advanced every day. Especially within news-media companies that tries to reach their customers with functional and promising algorithms. Natural language processing which is a very central part of companies that produce content can combine several models that compute bits and pieces of information about the text into a single model for a particular predictive goal. This research will use previously well explored natural language models to automatically extract information from text. The extracted information can be collected into a database which one can derive knowledge from. The news industries could then use this knowledge to analyse their supply and demand as well as creating summaries of their articles and paraphrase words to make it more understandable by certain target groups. While writing, there are different ways to express the message, while the gist of the text remains the same. Depending on the intended audience, an author can adapt the text with different formulation and terminology to ease the readers understanding of the text. The research problem is about finding this gist from any written text. This paper is trying to achieve this by extracting meta-data from the text. The algorithm developed in this paper is able to identify the gist of the sentence regardless of the grammatical structure and individual expression of each

David C. Wyld et al. (Eds): NATL, NETWORKS, Fuzzy - 2021
pp. 01-17, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.112001

author. With the information it would be possible for another algorithm or another person to adapt the text based on who is reading it. Adaptation of texts to different audiences could with this algorithm be dynamically automatized in the future. Therefore, the aim of this research is to extract meta-data in terms of entities and what relations these entities have with each other. This in turn can be captured automatically as the gist of the text. The text entities are words that identify object/things for example "house" or "cat" and relations are binding words between these entities such as "is" or "belongs to".

A knowledge graph one can quickly and flexibly query large data sets of entities together with their relations and sometimes even conclude new relations based on the imported information. The queries can answer what entities are connected and what relation they have with each other. From this information, possible answers can logically be derived by the algorithm. For example, if the goal is to know which colleagues are working with "Johannes Lindén", it is possible to query for entities that have a relation "colleague" with "Johannes Lindén". However, indirect relations and entities can also be found automatically. For instance, based on the sentence "Johannes Lindén works at Mid Sweden University", an indirect entity such as "A colleague" could also have the relation "works at" with the entity "Mid Sweden University". There is a significant number of indirect relations and entities which can be generated and one consequence is that it scales poorly in terms of storage and performance in relational databases. Instead, trained knowledge graph models can be used [1]. The data sets that these knowledge graph models are trained on are often generated from information extraction models. The information extraction algorithm developed in this research is called Generate Information Extraction (GenerateIE).

The goal of this research is to create an algorithm, GenerateIE, that extracts information from plain texts in multiple languages. A second goal is to combine a set of state-of-art algorithms to enhance the information extraction process of the GenerateIE algorithm. To evaluate these goals there are two metrics that will be considered. The first metric will be the accuracy of the number of correctly extracted data-points from the text. The second metric will be the intersection between a set of evaluation algorithms, e.g., a comparison of data-points that were found by one algorithm, but not by other algorithms. The novelty of this research is to use grammatical rules, common to multiple Germanic languages, to deduct which words are entities and relations. Another novelty is to extend the information extraction concept within area of news-media as well as reaching competitive accuracy with other state-of-the-art algorithms.

The remainder of this article is as follows: Section 2 presents relevant related work for this research. Section 3 presents our approach and the proposed model. Section 4 presents our evaluation thereof and the results. Finally, Section 6 presents our conclusions and our future work.

2. RELATED WORK

Information extraction is a difficult task, mostly because of complexity and variations in languages but also because there are few ways to evaluate the output [2, 3]. The output is generally a list of data-points. One data-point consists of three components, subject, relation and object, often referred to as a triple. Today's algorithms are capable to solve the information extraction task in English with either a greedy result with duplicated triples or missing entire sentences. They are today usually based on a fuzzy dataset, which is used to train a neural network model such as the OpenIE project [4]. A fuzzy dataset contains noise, and it is hard to determine the real accuracy and how well it performs for a certain use-case. The OpenIE algorithm has had several iterations of improvements over the years with contributors from for example Wu and Weld [4] as well as Angeli et al. [5]. Investigations of multi-lingual information extraction have been conducted by Claro et al. [6] which have a similar setup of supervised training. The OpenIE approaches all use a training data set for the information extraction itself and additional training on the dependent components such as dependency parsing and part-of-speech tagging as well.

Gashteovski et.al has developed an information extraction model called MinIE [7]. The MinIE algorithm is a combination of an older algorithm called ClauseIE developed by Del Corro et. al. [8] and aggressive information extraction optimization. The input of MinIE comes from the ClauseIE algorithm that suggests clauses to be considered as informative constituents of the input sentence. MinIE moves the constituents around until a potential relation is found.

There are things that complicate the matter of retrieving a sentence dependency tree and part-of-speech tags even more and that is if a sentence has entities in it that are spanning multiple words such as the cookie brand "Ben and Jerrys" or music band "Rolling Stones". The dependency parsing will not treat these words as entities but rather include them in the dependency tree as if they were separate. These entity problems require a trained named entity recognition model. Extracting information also complicates things when a sentence might refer to previous paragraphs or mentioned entities within such as he, she, it, that, them. A co-reference word model may deal with most of these sentence references which has been investigated by Clark et.al [9]. Since a sentence can be formulated in several different ways and still carry the same information, one consideration would be to try and simplify the sentence before using an information extraction model, a work conducted by Narayan and Gardent where the goal was just this to simplify the sentence [10].

Previous research is struggling to finding a good evaluation method since there are nearly no existing qualified data sets for the task of training such a model. Either a noisy labelled data set is used to estimate the performance of the algorithm, or the existing algorithm is compared with another in different ways. In the handbook of natural language processing, Alexander Clark et al.[11] are writing about different methods of dealing with evaluation when lacking a correct dataset within the NLP field. Alexander’s handbook among other articles summarizes the ways of evaluation into four different methods which are considered in this research [11, 12, 13].

1. Intrinsic evaluation: Manually tag a set of label and compute precision, recall and sometimes F-score.
2. Extrinsic evaluation: Use QA data set and match label with answer and question (for example WikiData→Wikipedia).
3. Laboratory evaluation: Letting people say if the predicted label is correct or incorrect.
4. Real world cases evaluation: Use people with expertise of which label should exist of sentences related to their field of expertise.

Intrinsic and laboratory evaluation requires some form of definition how to label the data points and although this was investigated, a decision was made to postpone this evaluation since the longer the sentences the more complex they became to label. The intrinsic evaluation is more complex and will therefore take longer time to label than the laboratory evaluation and the probability of error is also higher for the intrinsic evaluation. Extrinsic evaluation would work for a customer QA system. However, the news-media business case requires specific types of data which is not available for extrinsic evaluation at the time of writing. The real-world case evaluation re-quires experts within the English language and there are no openly accessible data sets for this purpose.

Google released a data source platform called GDELT [14] that stores billions of news metadata from all over the world, such a system could be used as valuable information to further enhance an IE algorithm. The system has the computer power to store and monitor world news on the internet from certain news sources, new events as well as events reaching as far back in time as 1979.Over 200 million events are recorded from over 240 countries and available for live requests. A similar system for crisis news is the Integrated Crisis Early Warning System (ICEWS). In 2013, a comparison between the GDELT and ICEWS was made that compared the popularity and scale of the two data sources. [15]

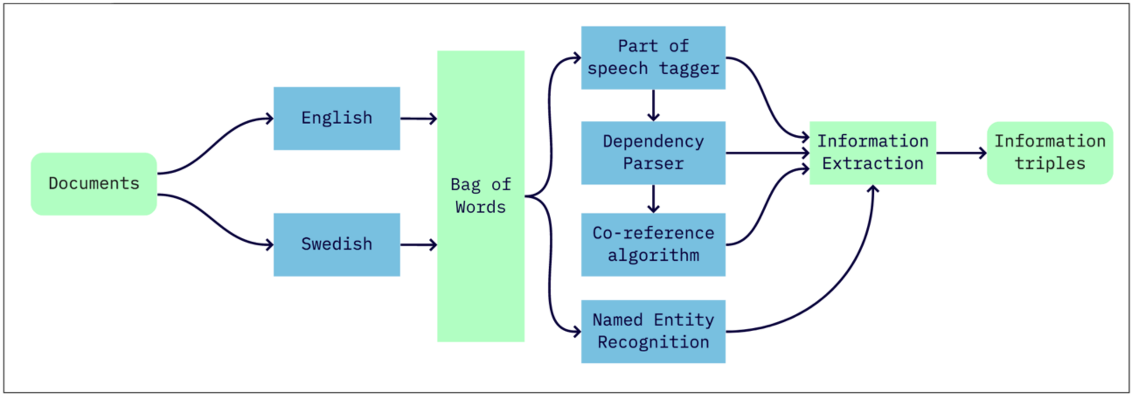


Figure 1. An overview of the proposed GenerateIE model. The input is an arbitrary text and the output is a list of data-points referred to as triples. A triple consists of a subject, relation and object components.

3. Method

The method of extracting information meta-data from texts used in this article is to propose a new algorithm called GenerateIE that is composed by several natural language processing models as well as a final algorithm that combines their predictions all together into a list of entity pairs and their relation, called a triple. Figure 1 shows an overview of how these natural language processing models are combined as well as the concept of the algorithm. Furthermore, the method section will explain the data sets, each natural language processing model, the final algorithm and how itis evaluated.

3.1. Dataset

The data set used when training the models of this research is Wikipedia articles selected by random sample with a total of 1000 articles. The data set is a good candidate since information can be extracted for different Germanic languages with texts of the same format and content. While preparing the data set some filtering was done for empty pages with no sentences as well as the format parameters within the articles written in XML syntax. The data set itself does not have any labels thus it is needed to manually evaluate the extracted triples by selecting reviewers. For the models requiring part-of-speech tags and dependencies another data set called Universal Dependencies [16] was used for the English language and a similar derived data set called "Talbanken" was used for the Swedish language [17, 18]. For the named entity recognition model, a pretrained model is fine-tuned with an additional entity type, relation, on WikiData data set. All relations in WikiData are extracted and a string match approach is used from the verbs written in Wikipedia articles.

3.2. Bag of Words

The first step in Figure 1 is a constructed one layered neural network model that will transform text paragraphs from the data set into several word vectors. This neural network is known as a bag of words algorithm. The word vectors are unique vectors that map the paragraph context associated with the word to a high dimensional vector space. A sigmoid activation function makes sure that the elements are bounded and by mapping the activation function according to Equation 1 it is ensured that the bounded values are within the interval -1 to 1. The vectors are constructed such that each element represents neighbouring words in a window. This way it is possible to relate a word by distance from another word [19]. The relation of two words can be obtained by computing the cosine similarity between their vector representations, see Equation 2. The cosine similarity will give a positive value when they are sharing similar contexts, a value close to zero when they have nothing in common, and a negative number when they appear in opposite contexts [19]. The vectors can also use common subtraction and addition operations to retrieve related word vectors, see Equation (3) for an example.

$$activation = \frac{2}{1 + \exp -X} - 1 \tag{1}$$

$$cosine\_similarity = \frac{vector_1 \cdot vector_2}{||vector_1||||vector_2||} \tag{2}$$

$$king - man + woman = queen \tag{3}$$

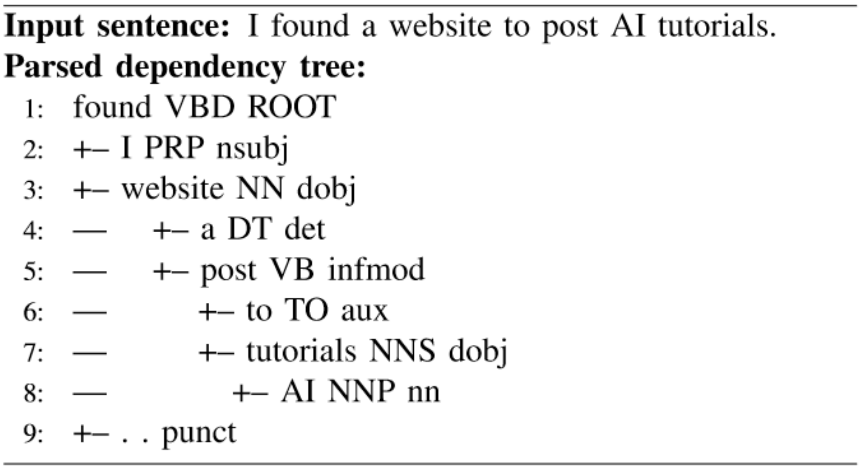
The Bag of Words algorithm takes an unstructured sequence of words forming a text sentence as input. The language of GenerateIE algorithm is for comparison reasons in English but Swedish language has also been evaluated in similar ways. Experiments in previously conducted research show that the continuous bag of words (CBOW) algorithm is reliable choice described by Mikolov et al [19].

3.3. Named Entity Recognition

To identify known entities, such as names of locations, organizations, people and more, a Named Entity Recognition (NER) model is used. Entities are identified by training a model that firstly recognizes in which context an entity usually exists, and secondly recognizes entity variances, i.e., multiple words can become an entity and certain words might be an entity in one context but not in another. A way to avoid that the dependency parser separates the potential entity words, a NER model is used to simplify prediction of the dependency tree and POS parser by replacing multiple word entities with similar single word entities. Different NER models were tried out, among them the model created by Finkel et.al. [20]. Among them, the GenerateIE performed best with a bidirectional BERT model transformer [21] by Google in terms of performance, number of entity types and a state-of-the-art research model. The supported types from the BERT model were Person, Organization, Time, Object, Event and Location. Since Time already is supported by the dependency parser, the entities used in this research were Person, Organization, and Location, Event and Object. Both the Swedish and English pre-trained models where fine-tuned on additional Wikipedia data while an additional entity type, Relation, was added. The Wikipedia dataset did not initially have any entity labels and therefore names of relations from WikiData was used by string matching the existing articles with the relation names.

3.4. Part-of-speech Tagger

The task of a Part-of-speech tagger (POS-tagger) model predicts a part-of-speech tag for each word in the input sentence. The tags could be Nouns, Verbs, Adverbs etc., and there is also a separate tag for special characters. Different languages have different number of tags. In total there are 55 tags in the English language, whereas the Swedish language has 21 tags. There are several algorithms that predict these tags such as SyntaxNet [22], and StanfordNLP [23]. In 2016 Google released a POS-tagger called SyntaxNet [24] with state-of-the-art performance, and one year later they announced an improved version [25]. In the experiments of GenerateIE the SyntaxNet algorithm is used to provide the English part-of-speech. The Swedish part-of-speech tagger is trained on a dataset from the resources mentioned in Nilson et al. [17]. The dataset is originally made by Jan Einarson's project is called Treebank [26, 18].



Input sentence: I found a website to post AI tutorials.
Parsed dependency tree:

```
1: found VBD ROOT
2: +– I PRP nsubj
3: +– website NN dobj
4: —    +– a DT det
5: —    +– post VB infmod
6: —        +– to TO aux
7: —        +– tutorials NNS dobj
8: —            +– AI NNP nn
9: +– . . punct
```

Figure 2. A part-of-speech example sentence parsed by SyntaxNet. The +– characters indicates a child path, followed by the word, part-of-speech tag and relation to patent word.

3.5. Dependency Parser

The task of a dependency parser is to break a sentence down to word dependencies. Each word in the sentence has a dependency to another word except for the root word. The dependencies between each word are named relations, for example "object to", "determinates" etc. Different dependency parsers provide different amount of named relations, e.g., in StanfordNLP [27] there are in total 47 named relations and in MaltParser [28] there are 65 named relations. In the GenerateIE algorithm, the StanfordNLP is used for the English language whereas the MaltParser is used for Swedish. See Figure 2 for an example of such a dependency tree together with the corresponding part-of-speech tags.

3.6. Co-reference Words (Word Linker)

A co-reference algorithm can be used to detect words in a sentence that also refers to other words in the context. For example, the sentence "John Doe went out for a break and he drank a cup of coffee", the co-reference algorithm is to identify that the word "he" refers to "John Doe". By linking these words together, the sentence dependency structure looks different and the extracted triples would make more sense if put into a knowledge graph. The knowledge graph would not contain the words he, she, it, them, they, etc. but instead they would be replaced by the original name of the identified entity. The co-reference models could be trained by using heuristic loss functions or reinforcement learning techniques [9]. The GenerateIE algorithm uses a co-reference model as a word linker to enhance the identification of entities. The mentions of a word will be linked together and replaced in the extracted information to enhance the value of each extracted word entity.

3.7. GenerateIE Algorithm

The GenerateIE algorithm will use the output of all previously mentioned algorithms (e.g., NER, POS-tagger, and dependency parser) and combine them into triples. The output of GenerateIE in Figure 1 is a set of triples T1, T2, ⋯, Tm. GenerateIE takes all the words and converts them into word-vectors using the bag of words algorithm. The triple extraction itself is rule based. That means that each word in a sentence is considered to be an entity and relation candidates. To narrow down the real entities and relations, GenerateIE utilize their part-of-speech tags as well as their word-dependencies. The part-of-speech tagger describes sentence parts as word classes. Dependency parsers work with set of grammatical rules in order to find relations between different word classes. This research uses the grammatical relations discovered by the dependency parser, and the entities found by the part-of-speech tagger, and then connects and reduces them into semantic entity-relation-entity triples, e.g., the elements to find the gist of a text.

Through the part-of-speech-tagger the likeliness of a word being a relation is increased by checking if the word is a verb. Similarly, the likeliness of a word being an entity is increased if the tag is a noun. Furthermore, neighbouring words of nouns are being concatenated depending on the grammatical rules found by the dependency parser. If the noun is bound to any word in the following list, the word would be considered to be part of the entity and concatenated with the noun:

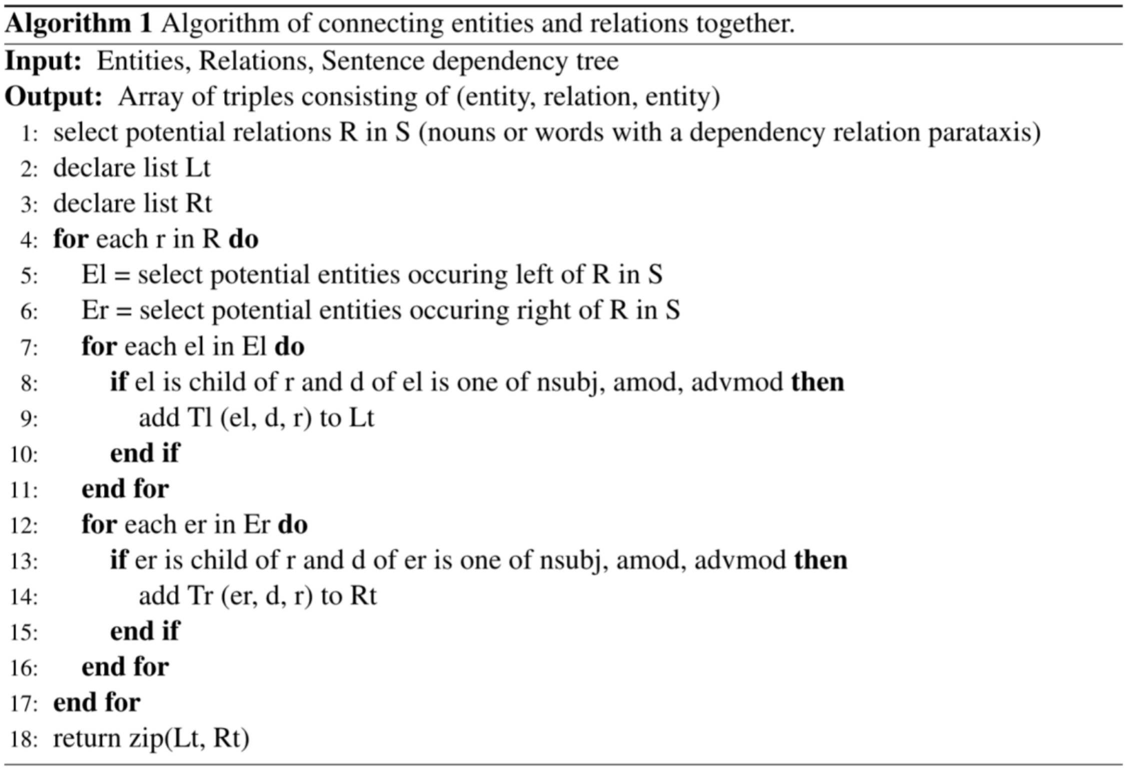
1. The noun is bound to an adjectival modifier (an adjective modifying the noun), for example, “the mother eats **red** meat".
2. The word is a number it is attached to the closest parent in the dependency tree.
3. The noun is bound to a nominal modifier, for example, "**Dr**. Andersson".
4. The noun is bound in a noun phrase as adverbial modifier, for example "I am 100 years **old**".
5. The noun is bound to a compound word, for example, "Let me borrow your **phone** book".

Similarly, the neighbouring words for a relation are being concatenated if the verb is bound to any of the following list, the word would be considered to be part of the relation and concatenated with the verb.

1. The verb is bound to an auxiliary word (a non-main verb), for example, "Meagan might have been lying"
2. The verb is bound to a copula word (link between subject to a subject complement), for example, "The sky is blue"

The found entities and relations are called soft entities and relations since they have yet not been connected into a triple. For entities the likeliness is made certain if the named entity recognition algorithm identifies one of the entities or their co-referenced word as an entity. For the relations the likeliness is made certain if the extended named entity recognition identifies them.

Once the relations and entities are defined, they are connected into triples. The method of doing this is to look at the dependency tree from the dependency parser and see if an entity is linked to another entity going through a relation upwards in the dependency tree towards the root element. Algorithm 1 shows the step-by-step instructions of connecting triples. More formally, parts of the algorithm can be expressed with set theory. For an input sentence Sin=w1, w2, ..., wn where wk is a word in the sequence of words of the sentence and |Sin|=n. The set of all words is defined as W={Sin}. All elements in W will be considered as relations and entities in T= (wk, wl, wm) where T is an ordered triple information in the sentence Sin. The first position of T is the subject, the second position is the relation and the third position is the object constrained to $0<k<l<m<=n$. The values of T are further limited to the condition wk $\in\{E\cap W\}$, wl$\in\{R\cap W\}$ and wm$\in\{E\cap W\}$ where the entity set E$\in$\{Noun,Conjuction\}and relation set R$\in$\{Verb\}. The subject and object words, wk and wm, should always be a children of the relation word wl. The child relations are denoted Tl= (wk,dl,wl) and Tr= (wm,dr,wl) for left and right entity respectively, where dl, dr $\in$\{subject,modification\} is the dependency relation between a pair of words in Sin. The retrieval of T from Sin is shown in Algorithm 1. Algorithm 1 can also be used for other Germanic languages like Swedish, given that there are part-of-speech and dependency parser models of the language we want to process. Since the models of OpenIE and MinIE are trained on an English data set, information extraction on Swedish was not possible at this point intime with these algorithms. The dependency parser rules used in this research are formulated for Germanic language structures and therefore a worse accuracy can be expected for other types of languages.



Algorithm 1 Algorithm of connecting entities and relations together.

Input: Entities, Relations, Sentence dependency tree

Output: Array of triples consisting of (entity, relation, entity)

```
select potential relations R in S (nouns or words with a dependency relation parataxis)
declare list Lt
declare list Rt
for each r in R do
    El = select potential entities occuring left of R in S
    Er = select potential entities occuring right of R in S
    for each el in El do
        if el is child of r and d of el is one of nsubj, amod, advmod then
            add Tl (el, d, r) to Lt
        end if
    end for
    for each er in Er do
        if er is child of r and d of er is one of nsubj, amod, advmod then
            add Tr (er, d, r) to Rt
        end if
    end for
end for
return zip(Lt, Rt)
```

There are two special rules about the entities. The first rule is about the named entity recognition algorithm. This will identify multi-word entities from the NER model (for example "Twenty century fox":organization) and make sure they are intact by verifying that an entity does not span multiple entities. After the first rule the second rule is applied. The rule checks whenever a co-referenced word is found that this word will be replaced with the identified entity.

3.8. Evaluation

The GenerateIE algorithm is evaluated using one quantitative and one manual evaluation for com-parison of the algorithms. The quantitative part consists of a comparison of the number of triples found in each algorithm. The reason is to check whether any algorithm is missing any triples that the other algorithms pick up and also to check for errors when an algorithm detects an unreasonable amount of them. The manual evaluation consists of a randomly sampled set of 100 documents were selected reviewers label triples by hand as being correct or incorrect. The labelling is used to compute an accuracy for the selected data set. The order of the triples generated by different algorithms have been randomized, so that reviewers would not recognize which algorithm generated which triples. This would then avoid rating one algorithm more favourably than another. As mentioned in Section 2, there are previous studies on four methods of manual data-labelling to con-sider in our research. A laboratory evaluation (method 3) was chosen over the intrinsic evaluation (method 1). Intrinsic evaluation is a more complex problem since identifying triples without help requires more knowledge than saying weather an already identified triple is correct or incorrect (as done in laboratory evaluation). Since no open accessible data set was found, the real-world cases evaluation (method 4) was not an option. The extrinsic evaluation (method 2) would be possible and should be evaluated, although for news-media the type of triples would need to be verified manually to ensure that the entities and relations actually existed in the text. Furthermore, there are no QA-labels for news-media articles that would be used in an extrinsic evaluation and therefore the laboratory evaluation (method 3) in Section 2 was chosen to evaluate the GenerateIE algorithm.

The label-reviewers, consisting of the members in the research team of five people, went through the extracted triples of all algorithms in a random order and determined from the text context if the data sets were correct or incorrect. The instructions[1] were handed to the label-reviewers of the subjective evaluation before they started labelling the dataset. The instructions consisted of an explanation of how to identify an entity and a relation, how to connect the objects into a triple, and how the labelling tool worked in detail. The explanation of how a triple is tagged is as follows.

Entity:

1. An entity could be a name of a location, person or organization
2. An entity could be I, you, he, she it, they, etc. referring to some definition in 1)
3. An entity can consist of several semantically connected words, for example "green apple" or “Adam Andersson" (note that in a sentence like "I like the green apple", both "green apple “and "apple" are valid entities for the relation "like", whereas "green" is a valid entity for the triple "apple is green")

Relation:

1. A relation consists of at least one verb connected to an entity.
2. A relation that exists along an adverb should be concatenated together to form the relation, relations which do not concatenate the adverb is incorrect (for example "he is running fast through the forest", "running fast" should be the relation)

Triple:

1. A triple can consist of an entity, relation, entity
2. A triple could consist of an entity, relation, adjective describing the entity (for example "The ball is blue")

Once the data set is labelled the algorithms can be compared by accuracy, but since this is a new data set an evaluation baseline is introduced in order to sort out algorithms that perform below this baseline as unsuitable for the information extraction task. A completely random triple picker

[1] Instructions available here:
https://docs.google.com/document/d/1PLUToV2drUlCTIXmLYeIWjpcHzhnLj6krA5MamEB8d0/edit

which considers all words to be entities and relations would have an accuracy of approximately zero. Therefore, the baseline has to be a "reasonable smart" random model that would pair each noun with each verb and consider all combinations of these as triples. Assuming there are one verb and two nouns per sentence the random model would yield an accuracy as shown in Equation4, but since the sentences should strive to be between 16-25 words [29] the accuracy is drastically reduced. Let's say that the baseline picks 3 words randomly from sentences, the accuracy would converge towards one fraction of 16 in best case and one fraction of 25 in worst case, see Equation6. Any algorithm that would have an accuracy above these base lines would be considered a candidate for the triple extraction task.

$$baseline\_accuracy\_3\_words = \frac{1}{5} = 0.2 = 20\% \tag{4}$$

$$baseline\_accuracy\_16\_words = \frac{1}{16} = 0.0625 = 6.25\% \tag{5}$$

$$baseline\_accuracy\_25\_words = \frac{1}{25} = 0.04 = 4\% \tag{6}$$

4. RESULTS

GenerateIE is quantitatively evaluated, comparing the number of extracted information triples, as well as manually evaluated, where a randomly selected sample set of the documents are labelled correctly or incorrectly to estimate the accuracy. The results are presented in the subsequent subsections.

4.1. Quantitative Evaluation

In total, 10000 Wikipedia articles were used to produce the results. Figure 3 shows the number of unique relations recorded for OpenIE, GenerateIE and MinIE after each document. Figure 4shows the matching between each pair of algorithms. It seems like the number of triples is not converging over time, but there are slightly less triples added by each article converging from an exponential increase towards a linear increase of newly found triples. The GenerateIE seem to find less relations than MinIE although finds more triples than MinIE in total as shown in Figure5. Most of the matched triples in Figure 4 are new (the lines are linear) and because multiple algorithms found the same triples, it motivates a higher probability certainty that the triples are correctly extracted triples.

Figure 5 shows a Venn diagram of the extracted triples of the two algorithms. The number of extracted triples of all algorithms are 398,408 triples, whereas 228,245 triples are only extracted by OpenIE, 25,401 triples only by MinIE and 129,966 triples only by GenerateIE. A few thousands of triples are found by all three algorithms. Extracting triples from even less articles was also tested but yielded the same ratios as Figure 5.

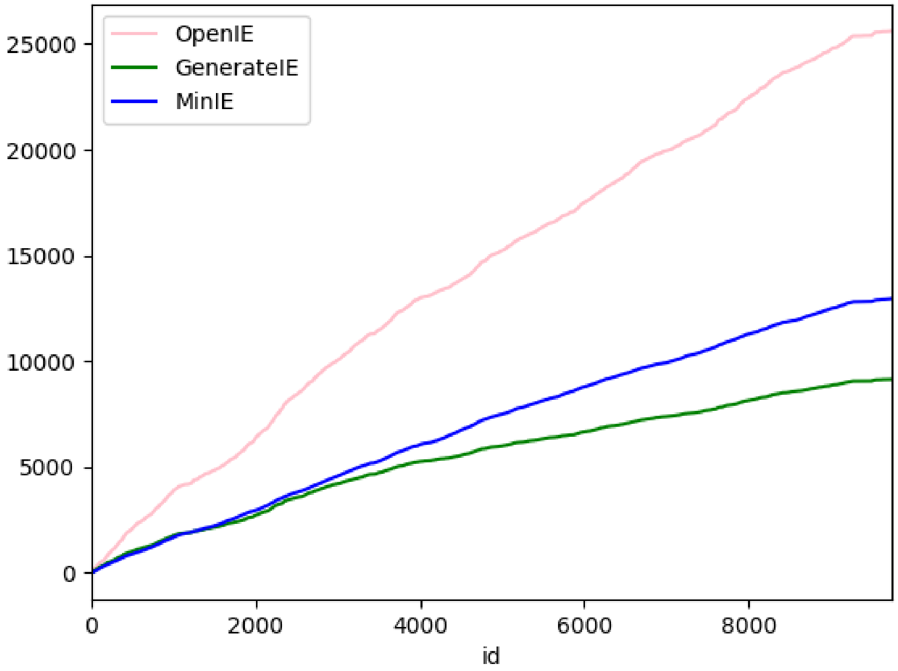


Figure 3. Total amount of unique relations extracted (y-axis) at a given iteration (x-axis)

4.2. Subjective Evaluation

The subjective results were produced by labelling the extracted information as correct or incorrect for each algorithm. In total 100 Wikipedia articles were labelled. Table 1 contains the numbers of triples identified by each algorithm together with how many of them where correct respectively incorrect. From these values, we compute the accuracy and standard deviation in the right-most column. The accuracy is the average number of correctly labelled triples shown in Equation 7. The reviewer's labels, correct and incorrect, could be represented in a stochastic variable X∈{0,1} where x=0 translates to incorrect label and x=1 translates to correct label. Since X could be either one or zero, the standard deviation uses the binomial distribution theorem about estimating the variance σ(X)2 in Equation 8 given ˆp from the previously computed accuracy in Equation 7.

$$\hat{p} = \frac{1}{n}\sum_{n} X \tag{7}$$

$$\sigma^2 = \frac{\hat{p}(1-\hat{p})}{n} \tag{8}$$

Figure 7 shows a bar plot of all extracted triples of the three algorithms and how many of those where correctly extracted. Figure 6 shows the accuracy of each algorithm and all possible combinations of them for the selected dataset. If the word linker is removed from GenerateIE

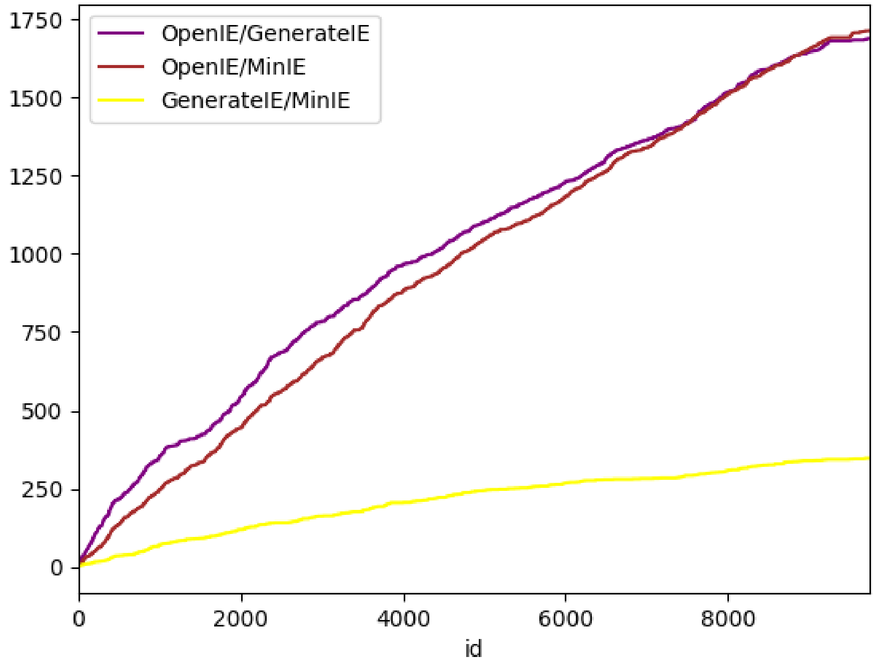


Figure 4. Total amount of unique relations extracted when matching pairs of algorithms (y-axis) at a given iteration (x-axis)

algorithm the accuracy is decreased to 31% while the standard deviation is also decreased to±2.3% as shown in Table 1. The total combination of the common triples of all algorithms are too small to evaluate further although computing the matched triples accuracy between OpenIE and GenerateIE yields83% as shown in Figure 6. The dataset is a subset of the quantitative evaluation with 100 articles, manually tagged triples with correct or incorrect. There are a total of 995 triples extracted by the algorithms, whereas 357 triples are only extracted by GenerateIE, 489 triples are only extracted by OpenIE and 51 only extracted by MinIE. There are a few overlaps where, a pair, or all algorithms have found the same triples. There are 98 correctly extracted triples by GenerateIE, 167 correctly extracted triples by OpenIE and 21 correctly extracted triples by MinIE. The overlap is slightly less when looking at the correctly extracted triples.

Based on this subjective evaluation, it seems like the OpenIE algorithm finds a lot of entities and relations that are not really classified as entities nor triples and it often suggests different mutations of the same entity which yields incorrect result, which may or may not be desired in an application using the algorithm. GenerateIE and MinIE are more conservative in suggesting triples and only suggest one triple combination for each identified subject and object pair. This impression is strengthened by Figure 3, where the line is steeper for OpenIE algorithm. The accuracy seems to be better in the MinIE and OpenIE algorithms compared to GenerateIE, but it is not significant at this point since it falls within the standard deviation interval.

Additionally, of the comparison a Swedish evaluation of the GenerateIE algorithm have been done. It has 79 triples less than the English evaluation of GenerateIE with an accuracy of 41% and a standard deviation of 2.8% as shown in Table 1. The GenerateIE algorithm evaluated on the Swedish dataset does have a high accuracy compared to the English dataset of the same algorithm.

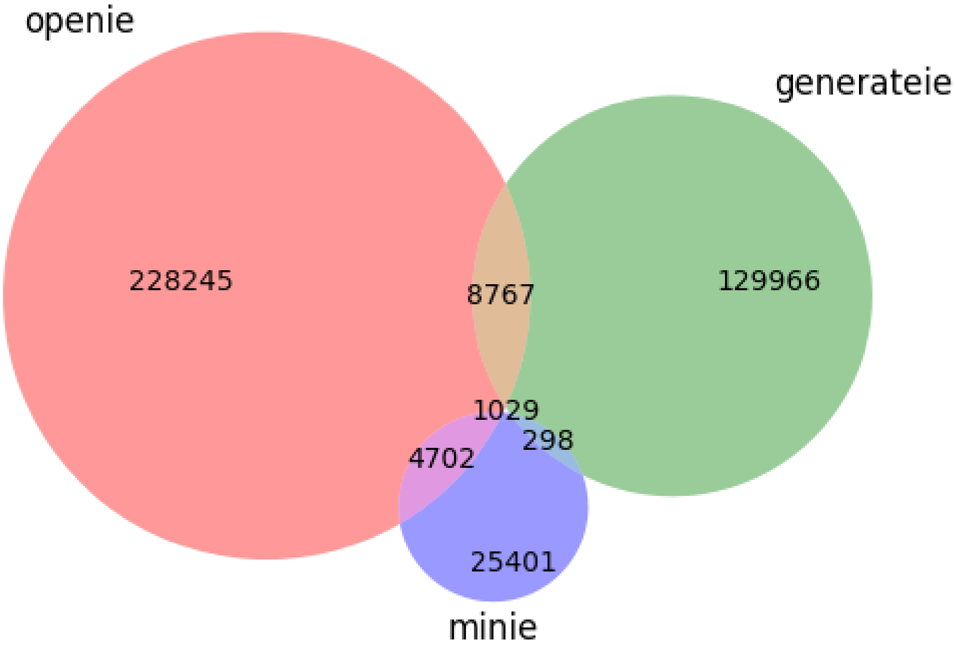


Figure 5. The triples extracted for the compared algorithms on 1000 articles

Table 1: The identified triples of OpenIE and GenerateIE and the result in accuracy and standard deviation when performing 10 cross fold validation

Source	Triples	Error	Correct	Accuracy	Std
All Combined	995	640	355	0.384	±0.016
OpenIE	533	333	200	0.375	±0.021
GenerateIE	394	272	122	0.310	±0.023
GenerateIE+WordLinker	394	272	122	0.378	±0.027
MinIE	68	35	33	0.485	±0.061
Overlap	3	0	3	1.0	0.0
GenerateIE (Swedish)	315	186	129	0.410	±0.028

5. Discussion

After the evaluation there are several aspects that should be highlighted in all the algorithms. The overlapped triples are very few and thus we cannot say very much about a combined accuracy in this article but presumably when all algorithms extract the same triple, it is likely that it is correct. The data-points are relatively few when quantifying the numbers - 100 articles and 995 triples in total - but it is more than enough to compute stable accuracy and standard deviations. There is a slight significance that GenerateIE is performing better with the word linker than without, and that MinIE outperforms all other algorithms with the 68 triples that where been found in the dataset, although that few triples mean that it is missing a lot of triples on almost half of the articles. With the GenerateIE word linker there are no significance between

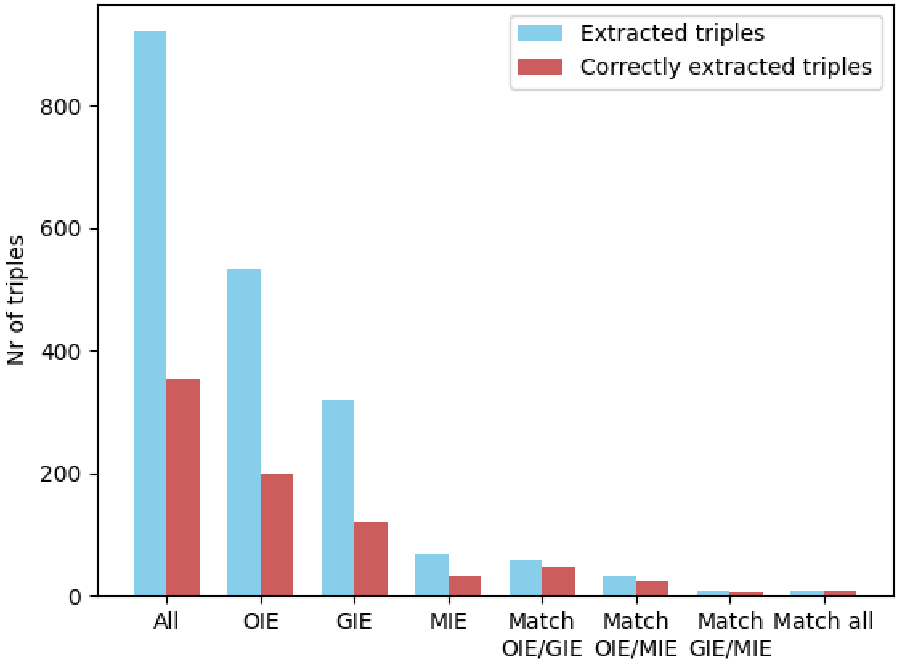


Figure 6. The number of triples found by each algorithm on 100 randomly selected articles from the entire dataset that was extracted and correctly extracted. To make room in the figure the naming was shortened (OIE=OpenIE, GIE=GenerateIE, MIE=MinIE)

OpenIE and GenerateIE. Furthermore, all algorithms are significantly higher accuracy than the baseline discussed in Section 3.8

The number of triples found by OpenIE is two times more than the GenerateIE algorithm in the quantitative evaluation the MinIE is even fewer. Only a small amount of the triples, 0.3%, are extracted by all three algorithms. The overlap is too small to give a just accuracy over all three algorithms, although adding common triples for OpenIE and GenerateIE we get an overlap of about3% and an accuracy of about 83%. The triples not found by any of the algorithms are unknown since we don't have a dataset tagged with triples in our evaluation. The lack of knowledge of total number of triples in a sentence might make the accuracy higher than it actually is e.g., the precision will be lower when including the unknown triples. Another important fact to consider when com-paring these algorithms is that the GenerateIE algorithm can handle multiple languages without any additional tweaks to the main algorithm as long as the part-of-speech and dependency parser exists for the language of the input text, the accuracy seem to be slightly higher of this evaluation even though the standard deviations overlap the OpenIE algorithm and the GenerateIE algorithm with word linker capabilities. It seems like the number of the unique found triples begin to in-crease linear but slightly decreases towards the end for both algorithms for each iteration of new documents we extract information from. The curve could be estimated as an O (n log n) function.

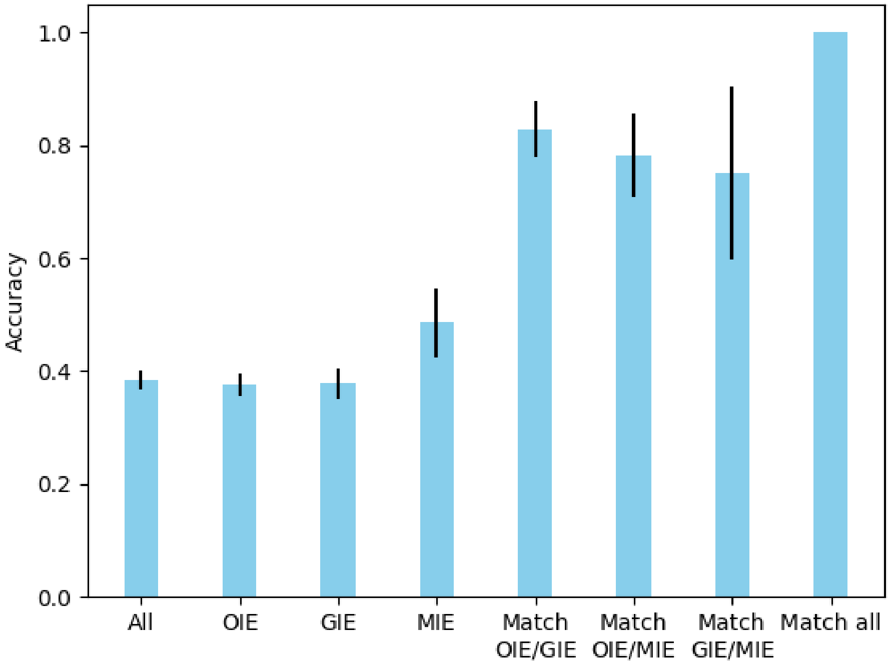


Figure 7. The accuracy of each algorithm on 100 randomly selected articles from the entire dataset. The black lines indicate the standard deviation of each evaluation. To make room in the figure the naming was shortened (OIE=OpenIE, GIE=GenerateIE, MIE=MinIE)

6. Conclusion

The goal of this research was to create the algorithm, GenerateIE, that combines existing algorithms to extract entity-relation-entity triples from plain texts to summarize the gist of it. The extraction can be done with multiple Germanic languages for texts in the news-media domain. A second goal was to combine a set of state-of-art algorithms to enhance the information extraction process. The GenerateIE algorithm can extract triples that will with 36% probability represent a partial gist of a sentence or paragraph. The algorithm has been compared and evaluated with two other algorithms, OpenIE and MinIE, which have the same goal. This evaluation shows that the GenerateIE algorithm performs slightly worse than the others in terms of the number of identified triplets, but in terms of accuracy it performs better than MinIE and worse than OpenIE. Additionally, GenerateIE gives the possibility to transfer the rules to different Germanic languages and thereby also make it possible to use in a multi-language approach. Even though the MinIE and OpenIE algorithms does not support Swedish language the evaluation of the GenerateIE algorithm on Swedish language shows that the accuracy is even higher than its English accuracy on a similar dataset. News-media companies will be able to use this algorithm to further analyse their con-tent. Further also tell what the readers are interested in reading on a much more detailed level than before for both Swedish and English texts. The impact of this work will also affect other domains within NLP such as how one can approach summarizing of text as well as deriving more knowledge from additional languages.

Regarding future work, a continued study is required to tweak the parameters of the algorithm to increase the number of identified triplets and the accuracy. An intrinsic evaluation should be performed to confirm the result in this paper for other languages than English and Swedish. The significance of non-found triples should be further evaluated because the evaluation in this research only reflects the accuracy of the found triples leaving an uncertainty in the false negatives affecting the recall score. It needs further to be determined how this research can be applied to different domains other than the news-media domain.

REFERENCES

[1] X. Han, S. Cao, X. Lv, Y. Lin, Z. Liu, M. Sun, and J. Li, "Openke: An open toolkit forknowledge embedding," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 139–144, 2018.

[2] A. Roy, Y. Park, T. Lee, and S. Pan, "Supervising unsupervised open information extraction models," 2020.

[3] G. Stanovsky, J. Michael, L. Zettlemoyer, and I. Dagan, "Supervised open information ex-traction," vol. 1, 2018.

[4] F. Wu and D. S. Weld, "Open information extraction using Wikipedia," in Proceedings of the48th annual meeting of the association for computational linguistics, pp. 118–127, Association for Computational Linguistics, 2010.

[5] G. Angeli, M. J. J. Premkumar, and C. 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), pp. 344–354, 2015.

[6] D. B. Claro, M. Souza, C. Castellã Xavier, and L. Oliveira, "Multilingual open information extraction: Challenges and opportunities, "Information, vol. 10, no. 7, p. 228, 2019.

[7] K. Gashteovski, R. Gemulla, and L. del Corro, "MinIE: Minimizing facts in open information extraction," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, (Copenhagen, Denmark), pp. 2630–2640, Association for Computational Linguistics, Sept. 2017.

[8] L. Del Corro and R. Gemulla, "Clausie: Clause-based open information extraction," in Proceedings of the 22nd International Conference on World Wide Web, WWW '13, (New York, NY, USA), p. 355–366, Association for Computing Machinery, 2013.

[9] K. Clark and C. D. Manning, "Deep reinforcement learning for mention-ranking coreference models," in Empirical Methods on Natural Language Processing, 2016.

[10] S. Narayan and C. Gardent, "Unsupervised sentence simplification using deep semantics," arXiv preprint arXiv:1507.08452, 2015.

[11] A. Clark, C. Fox, and S. Lappin, The handbook of computational linguistics and natural language processing. John Wiley & Sons, 2013.

[12] P. Paroubek, S. Chaudiron, and L. Hirschman, "Principles of Evaluation in Natural Language Processing," Traitement Automatique des Langues, vol. 48, pp. 7–31, May 2007.

[13] J. G. Smith and R. Tissing, "Using computational text classification for qualitative research and evaluation in extension., "Journal of Extension, vol. 56, no. 2, p. n2, 2018.

[14] K. Leetaru and P. A. Schrodt, "Gdelt: Global data on events, location, and tone, 1979–2012," in ISA Annual Convention, vol. 2, Citeseer, 2013.

15] M. D. Ward, A. Beger, J. Cutler, M. Dickenson, C. Dorff, and B. Radford, "Comparing GDELT and ICEWS event data, "Analysis, vol. 21, pp. 267–297, 2013.

[16] D. Zeman, M. Potthast, M. Straka, M. Popel, T. Dozat, P. Qi, C. Manning, T. Shi, F. G. Wu,X. Chen, Y. Cheng, A. Björkelund, A. Falenska, X. Yu, J. Kuhn, W. Che, J. Guo, Y. Wang,B. Zheng, H. Zhao, Y. Liu, D. Teng, T. Liu, K. Lim, T. Poibeau, M. Sato, H. Manabe, H. Noji,Y. Matsumoto, Ö. Kırnap, B. F. Önder, D. Yuret, J. Straková, C. Vania, X. Zhang, A. Lopez,J. Heinecke, M. Asadullah, J. Kanerva, J. Luotolahti, F. Ginter, Y. Kuan, P. Sofroniev,E. Schill, E. Hinrichs, D. Q. Nguyen, M. Dras, M. Johnson, X. Qian, Y. Liu, D. Vilares,C. Gómez-Rodríguez, L. Aufrant, G. Wisniewski, F. Yvon, S. D. Dumitrescu, T. Boro ̧s,D. Tufi ̧s, A. Das, A. Zaffar, S. Sarkar, H. Wang, H. Zhao, Z. Zhang, R. Hornby, C. Tay-lor, J. Park, M. de Lhoneux, Y. Shao, A. Basirat, E. Kiperwasser, S. Stymne, Y. Gold-berg, J. Nivre, B. K. Akku ̧s, H. Azizoglu, R. Cakici, C. Moor, P. Merlo, J. Henderson,H. Wang, T. Ji, Y. Wu, M. Lan, E. de la Clergerie, B. Sagot, D. Seddah, A. More, R. Tsarfaty,H. Kanayama, M. Muraoka, K. Yoshikawa, M. Garcia, and P. Gamallo, "CoNLL 2017

sharedtask system outputs," 2017. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[17] J. Nilsson and J. Hall, Reconstruction of the Swedish Treebank Talbanken. Matematiska och systemtekniska institutionen, 2005.

[18] J. Einarsson, "Talbankens talspråkskonkordans," Proc. of LREC, 1976.

[19] Q. V. Le and T. Mikolov, "Distributed Representations of Sentences and Documents.," in ICML, vol. 14, pp. 1188–1196, 2014.

[20] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in Proceedings of the 43rd annual meeting on association for computational linguistics, pp. 363–370, Association for Computational Linguistics, 2005.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[22] A. Voutilainen, "Part-of-speech tagging, "The Oxford handbook of computational linguistics, pp. 219–232, 2003.

[23] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of Tricks for Efficient Text Classification," arXiv preprint arXiv:1607.01759, 2016.

[24] D. Andor, C. Alberti, D. Weiss, A. Severyn, A. Presta, K. Ganchev, S. Petrov, and M. Collins, "Globally normalized transition-based neural networks," arXiv preprint arXiv:1603.06042,2016.

[25] C. Alberti, D. Andor, I. Bogatyy, M. Collins, D. Gillick, L. Kong, T. Koo, J. Ma, M. Omer-nick, S. Petrov, et al., "Syntaxnet models for the conll 2017 shared task," arXiv preprintarXiv:1703.04929, 2017.

[26] J. Einarsson, "Projektet talbanken. i: C platzack (utg), svenskans beskrivning 8, s76-96,"1974.

[27] S. Schuster and C. D. Manning, "Enhanced english universal dependencies: An improved representation for natural language understanding tasks," in Proceedings of the Tenth Inter-national Conference on Language Resources and Evaluation (LREC'16), pp. 2371–2378,2016.

[28] C. Bosco, S. Montemagni, A. Mazzei, V. Lombardo, A. Lenci, L. Lesmo, G. Attardi, M. Simi, A. Lavelli, J. Hall, et al., "Comparing the influence of different treebank annotationson dependency parsing," in Seventh conference on International Language Resources and Evaluation (LREC'10), pp. 1794–1801, European Language Resources Association (ELRA),2010.

[29] B. B. Kadayat and E. Eika, "Impact of sentence length on the readability of web for screen reader users," in Universal Access in Human-Computer Interaction. Design Approaches and Supporting Technologies (M. Antona and C. Stephanidis, eds.), (Cham), pp. 261–271, Springer International Publishing, 2020.