FREDA: Flexible Relation Extraction Data Annotation

Michael Strobl
University of Alberta
Edmonton, Canada
mstrobl@ualberta.ca

Amine Trabelsi
Université de Sherbrooke
Sherbrooke Québec, Canada
amine.trabelsi@usherbrooke.ca

Osmar Zaïane
University of Alberta
Edmonton, Canada
zaiane@ualberta.ca

ABSTRACT

To effectively train accurate Relation Extraction models, sufficient and properly labeled data is required. Adequately labeled data is difficult to obtain and annotating such data is a tricky undertaking. Previous works have shown that either accuracy has to be sacrificed or the task is extremely time-consuming, if done accurately. We are proposing an approach in order to produce high-quality datasets for the task of Relation Extraction quickly. Neural models, trained to do Relation Extraction on the created datasets, achieve very good results and generalize well to other datasets. In our study, we were able to annotate 10,022 sentences for 19 relations in a reasonable amount of time, and trained a commonly used baseline model for each relation.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**;

KEYWORDS

Datasets, Relation Extraction, Information Extraction, Information Retrieval

ACM Reference Format:
Michael Strobl, Amine Trabelsi, and Osmar Zaïane. 2023. FREDA: Flexible Relation Extraction Data Annotation. In *Proceedings of ACM SAC Conference (SAC'23).* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/xx.xxx/xxx_x

1 INTRODUCTION

The task of Relation Extraction (RE) is one of the major parts of Knowledge Base Population (KBP) [10], i.e. the augmentation of an existing Knowledge Base (KB). The main goal is to recognize relations, which are expressed between two entities mentioned in the same sentence or document. At present, this is usually achieved by a model based on neural networks, which is trained on, ideally, large amounts of labeled data. However, there are very few publicly available labeled datasets. When available, these are usually limited to specific relations, and often lack the relations that one is interested in. This leads to the question on how new datasets for RE can be created. Typically this is a tradeoff between time efficiency and accuracy. Through crowd-sourcing, large manually annotated datasets, such as the TAC Relation Extraction Dataset (TACRED) [25], can be created relatively quickly, although quality may be questionable. Recently, Alt et al. [1] gave some insights on TACRED, suggesting that more than 50% of the challenging examples, i.e. trained models make mistakes on, may need to be relabeled. On the other hand, carefully manually annotated datasets with multiple annotators, such as KnowledgeNet [15], are extremely time-consuming to create and therefore often not feasible. In this work, we propose an approach that makes it possible to create high-quality datasets with a moderate amount of time and effort.

To illustrate the difficulty of the annotation task for RE, consider the following example[1]:

> "**Melinda** began dating **Microsoft** CEO **Bill Gates** in 1987, after meeting **him** at a trade fair in **New York**."

Four entities are mentioned (**Melinda**, **Microsoft**, **Bill Gates** with his co-reference **him**, and finally **New York**). If the task is detecting the *ceo_of*-relation, the fact is explicit in the sentence. However, if the task is detecting the *spouse*-relation, one may indicate such relationship between Melinda and Bill Gates. While this might be true, it is only based on the annotator's knowledge and such annotation could improperly mislead a classifier and train it to perhaps associate "dating" with the *spouse*-relation.

Another possible annotation error can be illustrated by this example:

> "**Bill Gates** has received an honorary Doctorate from **Cambridge**."

An annotator confused about the meaning of **alma mater** could erroneously tag the *alma_mater*-relation between Gates and the university while Bill Gates never attended Cambridge. Although not always done (see TACRED [25]), this problem could be mitigated by using multiple annotators, as suggested by Alt et al. [1].

This leads to the question, how is it possible to create RE datasets, which: **(1)** are of high enough quality and validated; **(2)** are large enough to train effective models; **(3)** can be relatively easily extended with more relations; **(4)** can be quick to annotate and construct.

We believe these four conditions are essential for the creation of useful RE datasets.

Previous work encountered some difficulties to fulfill these conditions. Alt et al. [1] suggest that generating high quality and validated data (condition (1)) may not be met through crowd-sourcing or at least not if there are no measures in place that ensure data quality. Therefore, ideally, multiple annotators per example are necessary to ensure consistency. In addition, Rosenman et al. [19] suggested that all entities in a sentence need to be annotated in order to train models that generalize properly, which is not always done.

Creating large enough datasets (condition (2)) is not a trivial task and the required size of labeled datasets is usually unknown

SAC'23, March 27 –April 2, 2023, Tallinn, Estonia
© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-9517-5/23/03…$15.00
https://doi.org/xx.xxx/xxx_x

[1] From Melinda Gates' Wikipedia article https://en.wikipedia.org/wiki/Melinda_Gates

in advance. Indeed, labeling entities, co-references, and their relations to each other, even in a single sentence, can be complicated, depending on the relation in question, as well as time-consuming [15]. Moreover, existing datasets often contain a predetermined rigid set of common annotated relations like *located_in*, *founded* and *spouse*. Therefore, an easily extendable and flexible framework (condition (3)) is practical to gather data for new specific or uncommon relations. Enabling a quick construction (condition (4)) is not about the haste in annotation, which in turn could lead to errors, but more about the ease of annotation to avoid a repetitious tedious task. This ease of annotation is conducive to the collection of larger datasets (condition (2)).

Therefore, there is the need for an easy-to-use framework for sentence annotation for RE, with which the aforementioned conditions for such datasets can be met. These are the main contributions of this paper:

- We propose a framework, FREDA for Flexible Relation Extraction Data Annotation, which can be used to manually annotate sentences quickly and accurately. A simple procedure for sentence acquisition from a partially annotated Wikipedia-based corpus is provided to be able to create datasets for new relations.
- We provide a dataset with 10,022 sentences annotated for 19 relations (15 used by Mesquita et al. [15] and 4 new relations) with at least two annotators per sentence (third annotator for tie-breaking in case the first two disagree). Models trained on these sentences and their labels show a significant performance gain in F1 scores than previously reported results on common RE datasets, demonstrating that it is possible to obtain significantly better results when the annotations are of high quality.

The remainder of this article is structured as follows: Section 2 presents related work on RE dataset annotation. Section 3 describes our annotation approach. Section 4 delineates the model architecture. Section 5 details the evaluation procedure with our conclusions in Section 6.

2 RELATED WORK

The TAC Relation Extraction Dataset (TACRED) [25] is a large dataset which used Mechanical Turk crowd annotation with 41 relations and 106,264 examples. However, each example was only annotated by a single annotator and, as pointed out by Alt et al. [1], there is a large number of labeling errors misleading trained models. Although TACRED is still a popular dataset and widely used, e.g. by Joshi et al. [11] or Alt et al. [2], presumably since previous semi-automatic labelling approaches, such as text annotations with Distant Supervision (see Riedel et al. [18]), i.e. aligning sentences with facts from KBs only through matching entities, are even more error-prone. It seems to be important to double check annotations with more annotators, even though less text can be annotated that way.

In addition, Rosenman et al. [19] investigated the heuristics that a model trained on TACRED may learn to score high on the test set without solving the underlying problem: (1) Only 17.2% of the sentences in TACRED have more than a single pair of entities annotated. Therefore, in most sentences a model will only encounter a single pair of entities, for which it is asked to predict a relation. Instead of predicting a relationship for this pair, it may rather learn to predict whether a sentence expresses a certain relationship, ignoring the potential subject and object. This would lead to a high recall, but may also result in many false positives, leading to a low precision when tested.

(2) A classifier may predict a relation solely based on whether the types of entities in that relation are present in a sentence, especially for relations, which have a unique entity-type-pair, such as *per:religion* with PERSON as subject and RELIGION as object. The authors did a manual investigation of this relation with models trained on TACRED, often leading to false positives, if entities of type PERSON and RELIGION were present in sentences from Wikipedia, but the relation actually not expressed. One of their conclusions was that all entities in a sentence need to be annotated to lower the impact of these problems on the trained model, i.e. a model may generalize better to unseen data in this case.

KnowledgeNet [15] is a project aiming to manually annotate 100,000 facts for 100 properties, although at the time of publication 13,425 facts for 15 relations were available. Data annotation consists of the following steps:

1. Fetch sentences: Using T-REx [7], a system to align text and KB facts, to find sentences that could describe facts from Wikidata[2] and, in addition, sentences that contain certain keywords.
2. Mention Detection: Annotators are asked to highlight entity names.
3. Fact Classification: Pairs of mentions are classified as positive or negative for a relation.
4. Entity Linking: Linking mentioned entities to their corresponding Wikidata entity.

Each sentence is labeled by at least two annotators to ensure high data quality. The authors report an average of 3.9 minutes to annotate a single sentence by up to 3 annotators. In our approach, Entity Linking is not explicitly done. However, this step is only responsible for 28% of the time spent according to their study, and therefore a significant amount of time (about 2.8 minutes) is still needed to annotate a sentence.

There is also a variety of web-based annotation tools available, open-source tools, such as BRAT [22] or the INCEpTION project [13], as well as proprietary ones, such as Prodigy[3], which can be used for RE data annotation, among a variety of other NLP tasks. However, these tools are complicated to set up and use, making RE data annotation a time-consuming task, similar to [15].

In addition, it is important to note that there has been work conducted on few- or even zero-shot learning for RE in order to avoid the necessity of creating new datasets for relations, which do not appear in existing datasets. Han et al. [9] described a framework for few-shot RE and published the FewRel 1.0 dataset. It consists of 700 examples from Wikipedia for each of the 100 relations considered and was annotated by crowd workers. In contrast to our problem, the task for a model trained on FewRel 1.0 is to match an example, a sentence with two entities annotated, to a reference example from the set of examples of the true relation. Effectively, the goal is to

[2] https://www.wikidata.org
[3] https://prodi.gy/

rank relations and their reference examples. The relation corresponding to the best fitting example is selected. This allows the model to be trained on an arbitrary number of sentences (or even zero) and, hopefully, still select the correct relation through ranking it the highest. Based on FewRel 1.0, the FewRel 2.0 dataset [8] aims to fix two issues: (1) Adapt models to select relations from new domains and (2) avoid selecting a relation, if none of the available ones fit.

Even though the models used in [9] and [8] show promising results, Brody et al. [4] revealed similar issues with the FewRel framework to the ones found for TACRED by Rosenman et al. [19]. In particular, they found that models trained on FewRel 1.0 seem to heavily rely on entity types and adding training data for relations with similar entity types may mitigate this issue. Furthermore, the evaluation metrics used in FewRel ignores the fact that models may perform much better on some relations than others (in fact, they found a large gap between best and worst), potentially due to the aforementioned entity type issue. This, again, leads to the question how to annotate more data for more relations quickly and accurately.

3 DATA ANNOTATION WITH FREDA

Some works, such as Yu et al. [24], claim that most facts span multiple sentences since only Named Entities (NE) can be considered as subjects or objects. However, entities are usually referenced by pronouns or other co-references in the same sentence that expresses a property. Therefore, we claim that a sentence-based approach, like ours, is adequate in this case. The same one-sentence-based approach has also been applied by Mesquita et al. [15], even though a following Entity Linking step was used.

In general, manually creating labeled datasets for RE involves three tasks: (1) data acquisition, (2) data filtering (sentences in our case) and (3) the actual annotation task. Our approach for each of these tasks is described below.

3.1 Sentences from WEXEA

A major part of RE data annotation is selecting entities and their co-references, which could potentially be involved in a relation. Therefore, a corpus should be used with entities already labeled. This would only leave the decision on which entities are subject/object and whether a relation is expressed to the annotator. For example, a corpus for Named Entity Recognition (NER), such as the CONLL 2003 dataset [20], could be used. However, manually labeled corpora are usually limited in size and it is unlikely that they contain enough sentences that could be relevant for a specific set of relations.

To avoid this issue of potentially running out of text data, we are using sentences from WEXEA (Wikipedia EXhaustive Entity Annotations) [23]. WEXEA is an exhaustively annotated dataset derived from the English Wikipedia, which currently contains over 6,000,000 articles. Wikipedia already includes many hyperlinks to entities. However, some may still be missing. Indeed, Wikipedia editors are not encouraged to link the same entity twice, the entity an article is about, and widely known entities, whose link is obvious to the reader. However, WEXEA authors claim to capture all these missing links since it is easier than in, for example, news text, which typically does not contain any links to start with. Therefore, WEXEA can be used in our approach since it contains entity annotations already (including co-references), which can speed up the process. In addition, articles are already split into sentences.

WEXEA does not contain dates and times, but since some relations, e.g. *date_of_birth*, require these, we use the SUTime library [5], implemented within the CoreNLP tool [14], in order to add these as entities.

3.2 Sentence Filtering

If all available sentences are considered for annotation for each relation, the percentage of relevant sentences[4] is expected to be very low. This is an issue since the number of relevant sentences for each relation is presumably very imbalanced for a corpus without pre-selection, as it is the case for TACRED, for example. The by far most common relation in this dataset is *per:title* with 3,862 examples, whereas 37 out of 41 relations have less than 1,000 examples with 4 relations having even less than 100. Sentence filtering ensures that each relation has enough relevant examples to make sure a model can be trained on all relations properly. Therefore, we are using a similar approach as used by Mesquita et al. [15]:

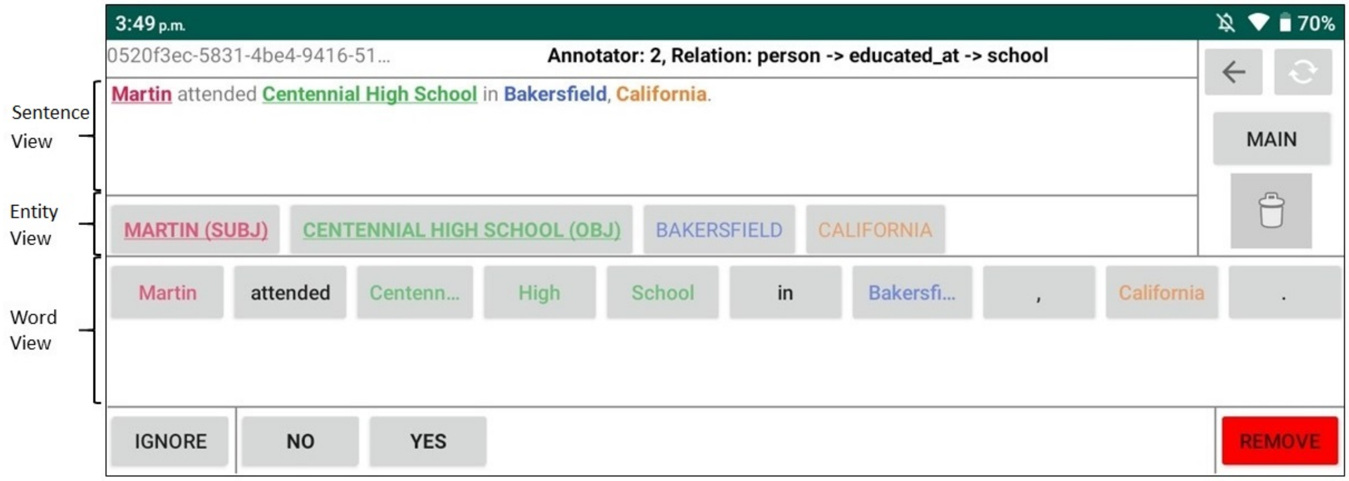
- Keywords: We define a set of keywords relevant to each relation, which are used to filter sentences. This is done for each relation separately. We chose relevant words for each relation including WordNet synonyms [16]. For example, for the *spouse*-relation we considered words such as "spouse", "wife", "husband", "married" or "wedding". The keywords used by Mesquita et al. [15] were not reported.
- Distant Supervision: Since it is difficult to define an exhaustive set of keywords, a Distant Supervision approach is used to find sentences not matching these keywords, but still containing entities, which are known to be related to each other with a relation of interest, e.g. the sentence mentioning Bill and Melinda Gates as seen in Section 1. Distant Supervision is typically used to add positive examples before training. In our case, we use distance supervision to add candidate sentences for annotation. If for a pair of entities a knowledge base indicates the existence of a relationship, we use these two entities as a query to find sentences as candidates for annotation for the stated relation. We are using DBpedia [3] as KB and extracted all entity pairs for each relation, which can be found in DBpedia.

Distant Supervision for a specific relation can only be used if this relation is part of a DBpedia, which is not the case for all relations we annotated sentences for. But if it was the case, up to 50% of the relevant sentences are extracted this way.

3.3 Sentence Annotation Task

Figure 1 provides an overview of the mobile annotation application of FREDA. The objective of the tool is to facilitate the annotation task and reduce the cognitive load for its users. This would lead to collecting annotations of decent quantities and high quality in a relatively short amount of time.

[4] It is somewhat subjective what a relevant sentence is for a specific relation. It could be a sentence which contains entities of both entity types participating in the relation or a sentence being somehow related to the relation topic-wise.



0520f3ec-5831-4be4-9416-51...

Annotator: 2, Relation: person -> educated_at -> school

Martin attended Centennial High School in Bakersfield, California.

Figure 1: Interface for RE data annotation FREDA. Relation considered here is "educated at" with "Martin" as subject and "Centennial High School" as object. The locations "Bakersfield" and "California" are not participating in the relation, but can be used for creating negative examples.

Given a particular relation, the tool initially provides a sentence with entity annotations highlighted in different colours in the *Sentence View*. These initial entity annotations are extracted and loaded by leveraging the WEXEA dataset and the SUTime library. The *Entity View* contains distinct entity buttons which are unique labels representing each entity annotated in the *Sentence View*. Thus, if multiple different mentions or co-references to the same entity occur in a sentence (*Sentence View*), they would all be represented by one entity button in the *Entity View*. The colour of that entity button and all corresponding mentions in the *Sentence View* would be the same. This would help reduce ambiguity and facilitate the decision making. Another practical advantage of the tool consists of reducing the indication of the subject (SUBJ) and the object (OBJ) entities in the relation to just a simple press of the corresponding entity buttons in the *Entity View*. In other words, it is not necessary to look through every single mention of an entity and indicate the role. Annotating only the representative entity button in the *Entity View* is sufficient.

The *Word View* contains buttons representing every single token in the sentence at hand. By using this view, conducting different editing operations becomes straightforward. For instance, new entities can be easily created through dragging and dropping one of the word buttons in the *Word View* up into the *Entity View*. Similarly, entities can be removed or fixed (e.g. adding a missing word) by just dragging and dropping. Web-based tools, e.g. BRAT[22], require the user to select spans of text using a computer mouse to create entities, requiring very precise (i.e. slow) moves.

In most sentences, editing operations using FREDA require a very minimal amount of time. Once done with editing, a user indicates whether the relation holds or not, i.e. a simple binary decision is made at the end. Sentences are considered for multiple relations as long as they meet the criteria outlined in Section 3.2. An annotator can also remove the sentence from the database (e.g. sentence is broken or list items) or ignore for the current relation.

3.4 Multiple Annotators

For such a complex task it is expected that a single annotator is not able to be accurate and consistent over hundreds of sentences. Alt et al. [1] show a detailed analysis of all the mistakes that are possible, especially for crowd-sourced annotation tasks, where presumably time matters more than accuracy. Therefore, similar to Mesquita et al. [15], our system relies on at least two annotators per sentence with a third annotator to break ties if the previous two disagree in their decision. The second annotator gets to see the entity annotations from the first annotator (and the third from the second). Note that only the entity annotations are carried over in this way. Thus, the final decision as well as indicating the subject and object entities is still every annotator's independent decision. But carrying over entity annotations saves time for subsequent annotators (also removed/ignored sentences are not shown to annotators thereafter). Therefore, the time spent for subsequent annotators is likely to be lower than for the first. In this work, the annotators were mainly researchers in the field of NLP.

4 MODEL ARCHITECTURES

We consider the best model architecture from Soares et al. [21], which is based on the BERT Transformer model [6] and showed good results on TACRED. Although instead of a multi-class classifier, as commonly used for models trained on TACRED, we altered the model to do binary classification. It is depicted in Figure 2. Apart from BERT's special tokens, two new tokens are introduced for the subject (entity start token *[ES]* and entity end token *[/ES]*) and for the object (entity start token *[EO]* and entity end token *[/EO]*), which are referred to as entity markers. The BERT embeddings of the start tokens of both entities concatenated are used as input of a classification layer with a sigmoid activation function, which makes a binary decision whether the relation of interest is expressed between the marked entities. Since sentences can express multiple

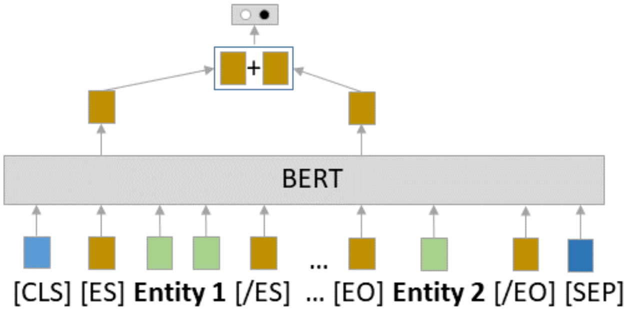


Figure 2: Model used for Relation Extraction from a sentence with annotated subject and object.

relations, even between the same entity-pair, independent classification decisions have to be made and one model per relation can be trained. Therefore, the input of the model for a given relation is a sentence with one entity mention marked as subject (with *[ES]* and *[/ES]*) and one entity mention marked as object (with *[EO]* and *[/EO]*).

This model architecture aims to solely recognize and decide whether or not the context of two entities suggests that the relation, the model was trained for, is expressed. There are no additional tasks such as NER or co-reference resolution that have to be learned and may influence the quality of the predictions. Therefore, this architecture should be suitable for finding out what performance is possible for the task of RE if accurately annotated and large enough datasets are available for training.

5 EVALUATION

In this Section, we provide a detailed evaluation of how models trained on datasets created with FREDA perform, how much data is required to reach a certain performance level for selected relations and how fast these annotations can be acquired.

5.1 Model Training and Dataset Statistics

We are using the previously mentioned model architecture outlined in Figure 2 with the cased large model of BERT. Learning rate of $5 * 10^{-6}$ (linear decay), Adam optimizer [12], batch size 32, 1-10 epochs (varies per relation; determined using 5-fold cross-validation). Test sets contain 10% of the whole dataset, one per relation.

Table 1 shows statistics of our datasets. 7 annotators annotated in total 10,022 sentences with >500 for 19 relation. 15 of these relations can be found in the KnowledgeNet dataset as well. In addition, we annotated 4 more common relations, which are also part of the schema.org ontology for persons[5]. *Positive responses* refers to the number of sentences, which were deemed as "expressing the relation", whereas *negative responses* correspond to sentences "not expressing the relation" for any pair of entities. From these positive responses, a number of *positive facts* can be extracted with a positive label, which can be used for model training. Often it is possible to extract multiple such facts for a single sentence since subjects and objects can be mentioned several times in the sentence and we may have different subjects or objects in the same sentence.

For instance, consider the following sentence:

> "**Princess Alberta** was the fourth daughter of **Queen Victoria** and **Prince Albert**."

Two positive facts can be extracted for the *child_of* relation:

- **Princess Alberta** *child_of* **Queen Victoria**
- **Princess Alberta** *child_of* **Prince Albert**

Facts with a negative label can be easily created by considering all other pairs of entities, which do not express the relation at hand. Therefore, four negative facts can be extracted from the previous sentence for the same relation[6]:

- **Queen Victoria** *child_of* **Princess Alberta**
- **Prince Albert** *child_of* **Princess Alberta**
- **Prince Albert** *child_of* **Queen Victoria**
- **Queen Victoria** *child_of* **Prince Albert**

Since there are typically more negative facts than positive ones, each training example is weighted in the loss function (binary cross-entropy loss) accordingly, in order to account for the class imbalance.

We calculated the inter-annotator agreement for the first and second annotator with Cohen's Kappa. Two annotators agree when they both consider that the relation in question is expressed (or inexistent) in the sentence at hand. Overall, the results can be considered as excellent with $\kappa = 0.85$ for all relations together. Although it ranges between 0.48 (*place_of_residence* and 0.96 (*date_of_birth*). Some relations require more discussion between annotators are therefore more difficult and time-consuming to annotate for humans, leading to more disagreement.

Statistics	Total
Relations	19
Sentences	10,022
Positive responses	5,371
Negative responses	4,651
Positive facts	11,160
Negative facts	232,678
Inter-annotator kappa	0.85

Table 1: Data statistics (Total): Number of sentences, number of yes- and no-responses, number of positive and negative facts extracted from these and inter-annotator kappa (between the first two annotators).

5.2 Test Results

We trained models on all datasets created with FREDA and tested them on the corresponding test sets as well as on unseen data provided by KnowledgeNet, which can be considered as high-quality.

The KnowledgeNet training data can be downloaded from their repository[7], which can be used for testing models trained on our datasets. Since this dataset does not contain exhaustive entity annotations (only entities participating in a specific relation are annotated), negative examples for testing can only be generated from

[5] https://schema.org/Person

[6] It is possible that these negative facts still express another relation, e.g. *parent* or *spouse*. But since the corresponding model is trained to do binary classification, they are considered as negative facts in this case.

[7] https://github.com/diffbot/knowledge-net

sentences expressing a relation. These are presumably the more challenging sentences since the model needs to figure out which entity is subject, which is object and which entities are neither. Also, entity-pairs for negative examples can be extracted through considering mentions of the same entity, which cannot be related to each other. For all relations, this results in 10,895 positive and 46,347 negative examples, compared to 11,160 positive and 232,678 negative examples for our datasets[8].

We are reporting Precision, Recall and F1 score on the FREDA and KnowledgeNet test sets in Table 2, broken up for each relation as well as **Interim** results for relations which can be found in the datasets from both approaches, and the **Total** which includes the four additional relations we annotated with FREDA.

The **Interim** F1 score of 0.86 on the FREDA test sets is relatively high overall. Even though it is not possible to directly compare this result against results of state-of-the-art models trained on the commonly used TACRED dataset, these latter usually achieve a significantly lower F1 score on TACRED[9]. Models trained on these datasets created with FREDA also show a similarly high F1 score on the KnowledgeNet dataset with 0.87. Although these datasets were created by different annotators and presumably similar, but still, in detail, different approaches for sentence filtering.

It is often not reported, but we can gain some insights on how such models perform on different relations. While relations, which do not leave a lot of room for interpretation, such as *date_of_birth*, *place_of_birth*, *child_of*, *spouse* or *sibling*, show a very high F1 score, others, such as *subsidiary_of* or *place_of_residence*, show significantly worse results. Cohen's Kappa for inter-annotator agreement for these two relations is quite low with 0.64 and 0.48, respectively, compared to the average of 0.85. Therefore, the datasets corresponding to these two relations can be considered as more challenging to train on, and thus more sentences may be needed.

5.3 Challenge RE dataset

Rosenman et al. [19] created a more challenging dataset based on 30 out of 41 TACRED relations, called Challenge RE (CRE). It is relatively balanced, i.e. the number of positive examples is similar to the number of negative examples, whereas TACRED is highly imbalanced. Furthermore, each annotated sentence contains at least two entity pairs that are compatible with the relation the sentence is annotated for, aiming to reveal models that learned to classify sentences rather than classifying entity pairs in a sentence, which would lead to high recall but low precision. Therefore, this dataset is considered to be more challenging than the TACRED test set. CRE was specifically created as a challenging dataset in order to test the generalization capabilities of models, for example, trained on TACRED.

We identified 7 relations in CRE that are fully compatible with 7 of FREDA's relations. , therefore the previously trained models on FREDA datasets can be used to be tested on the CRE dataset for these relations. In order to compare, we chose the KnowBERT-W+W model from Peters et al. [17], a knowledge-enhanced version of BERT through the integration of WordNet [16] and a subset of Wikipedia. It also uses entity markers for relation prediction and is trained on TACRED and shows state-of-the-art results on the TACRED test set.

Table 3 shows the results on the CRE dataset for both approaches. Overall, the F1 scores show that models based on FREDA often perform significantly better than KnowBERT-W+W, resulting in a higher total average. Another observation is that KnowBERT-W+W shows a very high recall compared to precision, which is expected due to the nature of TACRED and the resulting lack of generalization when tested on a more challenging dataset, such as CRE. FREDA's models, on the other hand, show a more balanced precision and recall, indicating that they pay more attention to which entity is subject and which is object, i.e. our datasets may lead to better generalization properties for models when trained on them, compared to TACRED. This also indicates that our sentence pre-selection step with keywords and distant supervision, which is important to end up with balanced datasets, is relatively general and does not necessarily only pre-select easy sentences, while still leading to a high F1 score when trained and tested on (see Table 2).

5.4 How many sentences do we need per relation?

TACRED contains a variety of relations with a high variance in the number of examples per relation (3,862 for *per:title* and only 33 for *org:dissolved*). However, typically only the overall performance of a model trained and tested on TACRED is reported in the literature, i.e. it is unknown how well these models perform on each relation and how many examples or sentences per relation are needed to reach a certain performance level.

We want to shed some light into the question of how many sentences have to be annotated per relation and how is it possible to find out whether more annotated sentences may be beneficial. Figure 3 shows the model performance on the FREDA test sets (same as used for the experiments reported in Table 2) for five different relations (*date_of_birth*, *spouse*, *educated_at*, *place_of_residence* and *subsidiary_of*), when trained on 100, 200, 300, 400 or all available sentences we annotated using FREDA and shuffled before sampling. These relations were selected since the models trained on the corresponding datasets show different performance levels (Table 2) as well as the inter-annotator agreement varies widely (Table 1).

The model corresponding to the least controversial relation among annotators (*date_of_birth*), i.e. the one with the highest Kappa, already shows stable performance after being trained on only 100 sentences. Models for the *spouse* and *educated_at* relations need slightly more sentences, but barely improve after being trained on more than 300. Whereas for the *place_of_residence* and *subsidiary_of* relations, even close to 500 sentences[10] seem to be insufficient to possibly get to a similarly high performance than the models for the other relations.

The Pearson Correlation Coefficient between model performance for each relation and the inter-annotator agreement is 0.75, i.e. both values are highly correlated. It can be concluded that if annotators often do not agree on annotations for certain relations, models have

[8] Each sentence can contain multiple positive and negative examples, each subject-object-combination is considered.

[9] https://paperswithcode.com/sota/relation-extraction-on-tacred

[10] As previously mentioned, we annotated at least 500 sentences per relation. However, 10% of these sentences are kept aside as test sets, i.e. the rest of the training sets may contain slightly less than 500 sentences.

	Datasets					
	FREDA (test)			**KnowledgeNet**		
Relation	**P**	**R**	**F1**	**P**	**R**	**F1**
date_of_birth (PER →DATE)	0.96	0.97	0.96	0.90	1.00	0.94
date_of_death (PER →DATE)	0.93	0.96	0.94	0.93	0.92	0.93
place_of_residence (PER →LOC)	0.71	0.76	0.73	0.86	0.74	0.79
place_of_birth (PER →LOC)	0.85	1.00	0.92	0.95	0.81	0.87
nationality (PER →LOC)	0.84	0.95	0.89	0.92	0.92	0.92
employee_or_member_of (PER →ORG)	0.68	0.91	0.78	0.95	0.82	0.88
educated_at (PER →ORG)	0.87	0.94	0.90	0.98	0.90	0.94
political_affiliation (PER →ORG)	0.96	1.00	0.98	0.90	0.90	0.90
child_of (PER →PER)	0.75	0.83	0.79	0.91	0.89	0.90
spouse (PER ↔ PER)	0.93	0.91	0.92	0.95	0.89	0.92
date_founded (PER →DATE)	0.83	0.95	0.89	0.94	0.88	0.91
headquarters (ORG →LOC)	0.80	0.84	0.82	0.94	0.86	0.90
subsidiary_of (ORG →ORG)	0.51	0.71	0.59	0.87	0.74	0.80
founded (PER →ORG)	0.72	0.94	0.82	0.49	0.82	0.61
ceo_of (PER →ORG)	0.81	0.89	0.85	0.94	0.91	0.93
Interim	0.83	0.90	0.86	0.88	0.86	0.87
award (PER →AWARD)	0.78	0.83	0.80	–	–	–
alma_mater (PER →ORG)	0.70	0.62	0.65	–	–	–
place_of_death (PER →LOC)	0.79	0.90	0.84	–	–	–
sibling (PER ↔ PER)	0.77	0.79	0.78	–	–	–
Total	0.82	0.89	0.85	–	–	–

Table 2: Test set results of the models trained on the FREDA training sets for each relation and both approaches. The last 4 relations are not part of KnowledgeNet's dataset, therefore the results are missing. Interim corresponds to the overall results for all 15 relations in both datasets. Total includes results on all relations in FREDA's test set.

	Models					
	FREDA			**KnowBERT-W+W**		
Relation	**P**	**R**	**F1**	**P**	**R**	**F1**
date_of_birth	0.96	0.93	**0.95**	0.67	0.99	0.80
date_of_death	0.74	0.78	**0.76**	0.61	0.74	0.67
educated_at	0.85	0.72	0.78	0.68	0.93	**0.79**
sibling	0.76	0.87	**0.81**	0.53	0.89	0.67
spouse	0.84	0.87	**0.85**	0.56	0.86	0.68
founded	0.86	0.53	0.66	0.82	0.76	**0.79**
date_founded	0.86	0.60	0.71	0.60	0.89	**0.72**
Total	0.83	0.76	**0.79**	0.63	0.87	0.73

Table 3: Challenge RE dataset test results. The previously trained models from FREDA were used as well as the KnowBERT-W+W model, trained on TACRED and showing state-of-the-art performance on the TACRED test set. The best F1 scores per relation and overall are in bold.

more difficulties to predict these relations as well, indicating that more data is needed. Whereas, if annotators barely ever disagree a relatively small amount of data is necessary.

5.5 Annotation Speed

Data annotation for RE can be prohibitively time-consuming. Therefore, approaches for quick dataset construction are essential in order to be able to easily extend existing datasets with more relations or create entirely new datasets. We asked two annotators, familiar with the task, to annotate sentences for the *spouse*-relation using the following approaches:

- **BRAT** [22]: BRAT is an open-source web-based tool for data annotation. Entities of different types can be annotated through selecting spans of text. Relations are annotated through connecting these entities.
- **FREDA (plain)**: In order to solely and fairly compare FREDA's annotation interface with BRAT, WEXEA entity annotations were removed for this approach.
- **FREDA**: Full framework, including annotations from WEXEA.

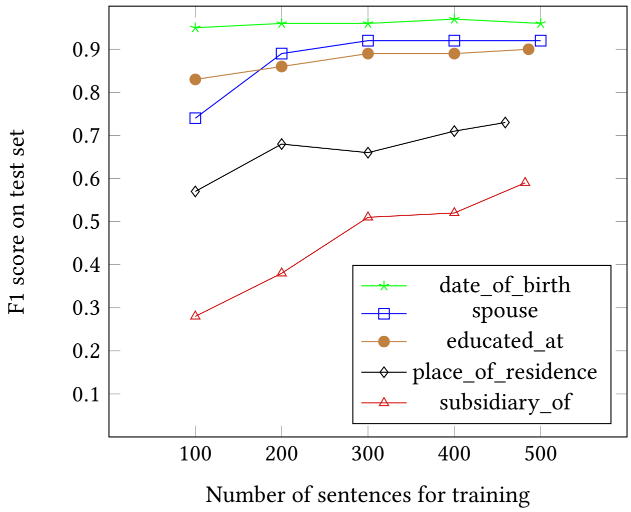


Figure 3: F1-score on test set for different training set sizes and a selection of relations.

	BRAT		FREDA (plain)		FREDA	
	sec.	**F1**	**sec.**	**F1**	**sec.**	**F1**
Annotator A	23.3	0.53	17.7	0.55	**9.2**	**0.69**
Annotator B	33.1	0.48	25.3	0.43	**12.4**	**0.56**

Table 4: Average annotation speed in seconds per sentence for each annotator, lower is better. A model was trained for each dataset and the F1 score on the CRE dataset for the *spouse*-relation as test set is reported. Best results per annotator are in bold.

Sentences were randomly selected from WEXEA. In order to keep the workload as similar as possible for each annotator and approach, all sentences contain exactly 25 words and a new unseen set of 100 sentences was used every time. In addition, the annotators were asked to only select entities of relevant types (*Person* in this case), which reduces the workload and the *spouse*-relation is not supposed to be applied to other types. However, WEXEA itself contains entity annotations of other types in which case annotators were not asked to remove entities for the approach **FREDA**.

Table 4 shows the average annotation speeds in seconds per sentence for both annotators and all three approaches. In general, annotation speeds vary significantly for each annotator since multiple steps are required and the speeds of each of them depend on individual abilities. The resulting datasets were used to train models, which were tested on the CRE dataset for the *spouse*-relation. Annotations with **FREDA** show the best F1 score for both annotators. Note that entities of other types are annotated as well using **FREDA**, while still keeping the average time to annotate a sentence low. This resulted in more examples for training and therefore better models when tested on CRE. Results for datasets from the other approaches are similar, suggesting annotation quality is similar[11].

For both annotators, the FREDA interface, represented through the approach **FREDA (plain)**, lead to a 24% increase in annotation speed compared to **BRAT** and another 48% to 51% increase for pre-annotated sentences from WEXEA, i.e. for the approach **FREDA** compared to **FREDA (plain)**, even though more entity types were annotated with **FREDA**. This should give an order of magnitude for the time required to annotated a sentence for a relation and how FREDA can help reduce the workload for annotators through its easy-to-use interface as well as using pre-annotated sentences.

6 CONCLUSION

Previous works on data annotation indicated either that large amounts of data are necessary in order to achieve moderate model performance (see TACRED [25]) or data annotation, if done carefully, is extremely time-consuming (see KnowledgeNet [15] or BRAT [22]). We showed that it is possible to create high-quality datasets for RE for a variety of relations, with a moderate amount of time and effort, using freely available text data from Wikipedia. The resulting models trained on these datasets showed state-of-the-art results for RE and are robust when tested on datasets from different annotators than they were trained on.

We hope that releasing FREDA to the public[12] will encourage the community to quickly create more annotated data for more relations, which would boost research for the tasks of RE and Knowledge Graph Population.

REFERENCES

[1] Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 1558–1569. https://doi.org/10.18653/v1/2020.acl-main.142

[2] Christoph Alt, Marc Hübner, and Leonhard Hennig. 2018. Improving Relation Extraction by Pre-trained Language Representations. In *Automated Knowledge Base Construction (AKBC)*.

[3] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference (ISWC'07/ASWC'07)*. Springer-Verlag, Berlin, Heidelberg, 722–735.

[4] Sam Brody, Sichao Wu, and Adrian Benton. 2021. Towards Realistic Few-Shot Relation Extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 5338–5345.

[5] Angel Chang and Christopher D. Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. 3735–3740.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[7] Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. 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)*.

[8] Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards More Challenging Few-Shot Relation Classification. 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)*. 6250–6255.

[9] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A Large-Scale Supervised Few-Shot Relation Classification

[11] All F1 scores in Table 4 are lower than reported in Table 3 since significantly less data was used for training.

[12] Annotated data, trained models and server and Android application code are publicly available: https://github.com/mjstrobl/FREDA

Dataset with State-of-the-Art Evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4803–4809.

[10] Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2010. Overview of the TAC2011 Knowledge Base Population Track. In *Third text analysis conference*.

[11] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics* 8 (2020), 64–77.

[12] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION Platform: Machine-Assisted and Knowledge-Oriented Interactive Annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, 5–9. http://tubiblio.ulb.tu-darmstadt.de/106270/ Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

[14] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010

[15] Filipe Mesquita, Matteo Cannaviccio, Jordan Schmidek, Paramita Mirza, and Denilson Barbosa. 2019. Knowledgenet: A benchmark dataset for knowledge base population. 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)*. 749–758.

[16] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[17] Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge Enhanced Contextual Word Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 43–54.

[18] Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 148–163.

[19] Shachar Rosenman, Alon Jacovi, and Yoav Goldberg. 2020. Exposing Shallow Heuristics of Relation Extraction Models with Challenge Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3702–3710.

[20] Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 142–147.

[21] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2895–2905.

[22] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. 102–107.

[23] Michael Strobl, Amine Trabelsi, and Osmar R Zaïane. 2020. WEXEA: Wikipedia EXhaustive Entity Annotation. In *Proceedings of The 12th Language Resources and Evaluation Conference*. 1951–1958.

[24] Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-Based Relation Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 4927–4940. https://doi.org/10.18653/v1/2020.acl-main.444

[25] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 35–45.