UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Software Engineering Curriculum

Vladyslav Umerenko

# Transformer-based Approach for Named Entity Recognition from Business News Articles for Company Detection

Master's Thesis (30 ECTS)

Supervisor: Rajesh Sharma, PhD

Tartu 2022

# Transformer-based Approach for Named Entity Recognition from Business News Articles for Company Detection

**Abstract:**
The advancements in mass media technologies made an enormous amount of news articles available online that mention private and public companies. Some institutions can benefit from the information that the texts contain about companies. However, the number of news articles is enormous, which makes manual text processing impossible. Consequently, researchers have always been looking for effective and efficient methods of company data detection from news articles, particularly company names. Among the possible applications of a company information retrieval is to detect mentions of companies and associate them with known companies in a database, namely company detection. This would allow keeping track of the activity of companies that may be crucial for private and public companies that provide services of company activity analysis, systematic traders and other bodies that depend on the monitoring of company activity. One of the approaches for company name detection is Named Entity recognition (NER), which is a task that aims to locate and classify proper nouns into a defined set of categories. While the traditional approaches of NER involve the use of rule-based, word-based and sequence modelling models, such as Long-Short Term Memory (LSTM) and Conditional Random Fields (CRF), the recent developments in Deep Learning enable to employ Transformer-based models in a combination with the sequence modelling to produce a state-of-the-art performance of NER. The research starts with the evaluation of a simple approach to company detection. Then some word-based models are assessed to estimate the necessity of more sophisticated models. Next, the models based on Transformer, namely RoBERTa-base, in a combination with the transition-based algorithm for NER are presented. Finally, the named entities that are predicted on news articles are employed in the company detection task. The experiments have shown that the exploitation of Transformer-based models results in a significant improvement of both Named Entity recognition and company detection tasks (two-fold improvement). In addition, the presented approach successfully solves the task of open-world company name recognition that can be used to expand the existing database of companies.

**Keywords:**
Named Entity recognition, company detection, Transformer, Machine Learning, RoBERTa, GloVe, Elasticsearch, spaCy

**CERCS:**
P170 Computer science, numerical analysis, systems, control

## Äriuudistes ettevõtete kindlaks tegemine kasutades trafopõhist lähenemist nimeüksuste tuvastamiseks

**Lühikokkuvõte:**
Massimeedia tehnoloogiate edusammud on teinud internetis kättesaadavaks tohutu suure hulga uudisteartikleid, mis mainivad nii era kui avaliku sektori ettevõtteid. Sellest informatsioonist võiksid kasu saada mitmed asutused. Kuna antud artiklite hulk on tohutu, siis manuaalne tekstitöötlus on võimatu. Seetõttu on teadlased alati otsinud tõhusamaid viise, kuidas tuvastada uudisteartiklitest ettevõtete andmeid, eriti just nende nimesid. Võimalike rakenduste seas on teabeotsing ettevõtte mainimisel ning nende seostamine juba andmebaasis olevate teadaolevate ettevõtetega. See võimaldaks jälgida ettevõtete tegevusi, mis on ülioluline ettevõtete tegevusanalüüsi teenuse osutajatele, süstemaatilistele kauplejatele ning mitmetele teistele osapooltele, kes sõltuvad ettevõtete tegevuse monitoorimisest. Üks võimalik lähenemisviise ettevõtete nimede tuvastamiseks on Nimeüksuste tuvastamine (NER), mis üritab leida ja klassifitseerida sobivad nimisõnad kindlaks määratud kategooriatesse. Kui traditsioonilised NER'i lähenemisviisid sisaldavad reeglitepõhist, sõnapõhist ning järjestuste modelleerimise mudeleid nagu pikk lühiajaline mälu (LSTM) ja tingimuslik juhuslik väli (CRF), siis hiljutised arengud sügavõppes võimaldavad tipptasemel NER'i loomiseks rakendada kombinatsiooni trafopõhistest mudelitest ning järjestuste modeleerimisest. Käesolev uurimus hindab kõigepealt lihtsaid ettevõte tuvastamise lähenemisi. Seejärel vaadeldakse mõningaid sõnapõhiseid mudeleid, et hinnata keerukamate mudelite vajadust. Järgmisena esitletakse kooslust trafopõhisest mudelist, täpsemalt RoBERTa-base'ist, ja järjestuste-põhisest algoritmist. Ning viimasena pannakse uudisteartiklitest saadud nimeüksused proovile ettevõtete tuvastamisel. Katsed on näidanud, et trafodel põhinevad mudelid on märgatavalt edukamad nii nimeüksuste leidmisel kui ettevõtete tuvastamisel (kahekordne täiustamine). Lisaks lahendab esitletud lähenemine edukalt ettevõtete nimede tuvastamise, mida saab kasutada olemasoleva ettevõtete andmebaasi laiendamiseks.

**Võtmesõnad:**
Nimega üksuste tuvastamine, ettevõtte tuvastamine, Transformer, masinõppimine, RoBERTa, GloVe, Elasticsearch, spaCy

**CERCS:**
P170 Arvutiteadus, arvanalüüs, süsteemid, kontroll

# Contents

# 1 Introduction

Nowadays, news articles describing the activity of companies are easily accessible. Such entities as private and public companies, anti-monopoly regulators and systematic traders may benefit from information about companies available in the news articles. However, the amount of news articles mentioning public and private companies is colossal to be processed manually. Therefore, it is crucial to develop efficient and effective methods for the automatic extraction of company information. One of the possible applications of such a method is to detect mentions of companies and associate them with particular known companies in a database. Such an approach would allow monitoring global news related to the companies of interest. The Named Entity recognition (NER) is one of the Information Retrieval (IR) tasks that can facilitate the extraction of company-related information, particularly company names.


Figure 1.1. Example of NER tagging with a company name entity.

The Named Entity recognition is a task of identification of proper nouns as belonging to particular categories of Named Entities (NE) [39]. It is one of the methods for company name extraction from texts. The term "Named Entity" denotes a real-world object that can be identified by a name, such as companies. Figure 1.1 represents an example of NER tagging on a real-world sentence. In general, the approaches for NER can be divided into two categories. Some early works on NER involve the application of rules, gazetteers and a search engine that applies the rules and the lexicons to the text [39, 21, 1, 51]. One of the disadvantages of such a method is that the rules are either handcrafted by humans or bootstrapped from a few examples of handicrafted features. In addition, gazetteers contain only a limited set of nouns, which would not be beneficial for company name entity recognition, since new companies emerge frequently nowadays. The second approach for NER applies statistical methods to produce probabilistic representations of the training data [23, 22, 38, 29]. The statistical modelling of texts generally produces a greater performance of NER models. In addition, the latest developments in Machine Learning (ML) and Deep Learning (DL) achieve state-of-the-art performance on Named Entity recognition task [62, 67, 70]. The detection of company names is the process of identifying the mentions of companies in texts. Specifically, it is crucial to recognize company name boundaries and correctly classify words belonging to the company name class in order to be able to associate the detected companies with real-world companies. This way, the Name Entity recognition approach can facilitate this task efficiently.

This thesis presents a Transformer-based model for company name entity recognition. The Transformer is a neural network architecture for language understanding that adopts

the mechanism of self-attention. The Transformer encoder adopts a fully-connected self-attention structure to model the long-range context, which is the weakness of Recurrent Neural networks (RNN) [69]. The self-attention enables to learn the weighted significance of words relatively to other words in a sentence so that a language model learns to focus more on important words rather than irrelevant [64]. It is shown that the application of context-based models, such as the presented, improves the boundary detection and entity classification performance significantly, compared to the word-based models, such as WordNet and GloVe. This work is conducted in a partnership with a private company [1] that is intended to use the developments in their product.

The goal of this work is to answer the following questions:

- **RQ1**: How can business news articles be associated with known companies? In other words, given a database of known companies, associate the articles with particular records in a company database. To illustrate, let's say there is a database containing such records mapped to identifiers (ID) as $(Facebook, 0001)$, $(WhatsApp, 0002)$ and $(Instagram, 0003)$. There is a set of news articles mapped to identifiers, i.e. $("On\ Monday...", 1001)$, $("Senate\ approved...", 1002)$ and $("Stock\ price\ of...", 1003)$. The system output would be a set of tuples such that each company ID is associated with a news article ID, such as $(0001, 1001)$, $(0002, 1002)$, $(0003, 1003)$.

- **RQ2**: How can company names be recognized from news articles? This question investigates how such entities that belong to the company name class can be predicted, given a set of news articles. For instance, given an article "Facebook, owner of Instagram and WhatsApp, reported a whopping 2.5 billion daily users on average in September across its platforms", the recognized company name entities would be "Facebook", "Instagram" and "WhatsApp". The answer to this research questions facilitated the answer to the first question through a company name recognition technique.

The thesis is organized as follows. Chapter 2 reviews previous developments in the NER task and summarises the difference between this thesis and the previous works. Then, Chapter 3 describes two datasets involved in this research. Chapter 4 introduces the research methodology step-by-step. After that Chapter 5 reports the approach for company detection that serves as a point of comparison with the application of NER to the same task. Next, Chapter 6 presents the company name entity recognition approach that investigates the exploitation of word-based and context-based (Transformer-based) methods and the application of NER in the company detection task. Finally, Chapter 7 concludes and summarizes the outcomes of experiments and suggests future directions for further research.

[1] This company is located in Eastern Europe

# 2 Related work

## 2.1 Background

The Named Entity recognition task was coined for the Sixth Message Understanding Conference (MUC-6). It defined the task as the identification of the names of all the people, organizations, and geographic locations in a text [25]. Obviously, it can be extended to other Name Entity types as well. Since then a great number of approaches have been developed from the application of rule-based grammars with statistical models to neural architectures such as bidirectional Long-Short Term Memory (Bi-LSTM) with Conditional Random Fields (CRF).

## 2.2 Traditional methods for NER

[39] evaluated the direct lookup of entities from the list of known persons, locations and organizations first. While this approach yielded reasonably good results on the location type, it didn't work well for the person and organization entities. After that, they combined contextual grammar rules with probabilistic partial match generating all possible partial orders of words, which is then fed to a pre-trained maximum entropy model. [27] explore a NER system based on Support Vector Machine (SVM) [11]. It is a machine learning algorithm for binary classification that learns to separate two classes with maximum margin. Since the multi-class application of SVM requires a one-versus-all training strategy, the researchers find SVM models inefficient and propose to remove useless features and train all the sub-models concurrently. [10] first proposed an architecture involving temporal Convolutional Neural Networks (CNN) over a sequence of words. In order to avoid task-specific engineering and man-made features carefully optimized for each of the NLP tasks the designed system learns internal representations on an extensive set of unlabeled data. Then a well-performing tagging system is developed from the learned representations. [54] investigates the capabilities of Conditional Random Field to predict biomedical Named Entities. CRF is an undirected graphical model conditioned on observation sequences, which enables to model dependencies between observations [65]. He combines CRF with such linguistic features as orthography and semantics to show that the use of semantic lexicons does not affect performance significantly. While the use of simple semantic features advances the performances to the near state-of-the-art values.

In [33] researchers combine bidirectional Long-Short Term Memory model with Conditional Random Field. Recurrent Neural Networks (RNN) is a family of artificial neural networks that operate on sequential input [56]. Such networks are prone to the exploding gradients problem that leads to large updates of model parameters as well as to a bias towards the latest input data. The LSTM model is an improvement of RNN models that resolves the described issues by the introduction of a hidden state. Further

developments lead to a combination of Bi-LSTM with CNN [8]. Such a combination learned both character and word-level features. While the recurrent layer transformed word-level features into named entity scores, the CNN layer extracted character-level features, in particular, character-level embeddings. Another approach that leverages a combination of CNN and LSTM uses two CNN layers connected with an LSTM layer [55]. The first CNN layer learns a character-level encoder, while the second learns a word-level encoder, which is then used to build a tag decoder with an LSTM. In addition, they take advantage of the active learning approach that aims to target a more informative set of examples in contrast to supervised learning, which requires drawing randomly selected examples.

## 2.3 Transformers for NER

Nevertheless, the recent advancements in Deep Learning for Natural Language Processing (NLP) have shown a significant improvement in terms of the development of general language models capable of solving different tasks. Particularly, it has been shown that the application of pre-training Transformer models for Named Entity recognition allows achieving state-of-the-art performance [67, 70].

[69] investigate a model that involves Transformer as a character-level encoder, where character-level features are concatenated to produce word-level features. Here CRF serves as a decoder layer. They claim the Transformer encoder performing poorly on the NER task, thus proposing a modification to the original Attention algorithm: direction and distance-aware attention and un-scaled dot product attention. In [2] researchers tune multi-lingual Bidirectional Encoding Representations from Transformers (BERT) with an adapted word-level CRF layer on top achieving improvement over Bi-LSTM-CRF model proposed by [33]. [30] present MT-BioNER — multi-task Transformer-based NER architecture to overcome complexity of new domain support and a lack of training data to tune a model for new domain, particularly, biomedical data. [34] suggest a multi-task Transformer-based model solving Question Answering (QA), Natural Language Inference (NLI) and Recognizing Question Entailment (RQE) tasks. They combine NLI task with NER task to achieve accuracy improvement. The researchers impelented a system capable of solving all 3 tasks with minimal changes thanks to available pre-trained Transformer parameters significantly reducing the training time necessary to produce a language representation model. The first end-to-end Transformer-based system was proposed performing handwriting recognition and NER on paragraph-level outperforming previous IEHHR Competition systems [48]. [52] employ fine-tuned Transformer model to learn document-level features and show that such an approach improves the overall $F_1$-score and fine-tuning outperforms the feature-based LSTM-CRF. In [62] researchers explore state-of-the-art models for parsing clinical trial eligibility criteria showing that Transformer-based models pre-trained with relevant clinical corpora can improve their performance. [67] exploit four pre-training Transformer-based NER

models, such as BERT, ERNIE, ERNIE2.0-tiny, RoBERTa, with a Fully-Connected (FC) and CRF layers on top of the models. They reveal a significant improvement of recognition with RoBERTa outperforming the rest of the models. The new state-of-the-art NER model, XLNet-BiLSTM-CRF, is produced adopting XLNet Transformer model [72] that outperforms other BERT-based models [70].

## 2.4 Summary

Most of the modern researches engage a combination of Transformer, Bi-LSTM, CNN and CRF layers in different orders to achieve near state-of-the-art results for NER task. Consequently, this approach looks promising for the recognition of company names from business news articles. The complexity of data collection for training company name entity recognition spans across the scale of the number of business news articles available. Thus, it is possible to obtain only a limited collection of news articles mentioning companies. Apart from that, the complexity of company NER lies in the company name morphology. Commonly, company names might not be included in the language lexicon. Thus, it is problematic to utilize dictionary-based methods. Consequently, it is vital to develop company NER systems that would be able to produce satisfactory performance on a deficient quantity of articles not relying on a dictionary to be used in real-world products. Some other challenges of NER include the detection of named entity boundaries as well as the recognition of named entity categories, i.e. belonging to a particular class [41]. Therefore, this work presents a RoBERTa-based model for company name entity recognition that employs a Transformer layer for feature extraction and a transition-based approach with a fully-connected layer as a classification layer for Named Entity recognition [66]. While other methods employ multi-layer LSTM and CNN models, the approach presented in this thesis employs a more space-efficient transition-based method with a single fully-connected layer for NER. Such an approach provides a good balance between accuracy, efficiency and adaptability.

# 3 Datasets

## 3.1 Collected datasets

**Dataset of news articles:** The dataset is collected by a partner company that this work is conducted with. The news articles are scraped from 10 news websites focused on business news and selected from categories relevant to the global public procurement topics in the English language. The dataset consists of 802,587 news articles from 2005 to 2021 years. The distribution of news articles per news source is displayed in Figure 3.1.

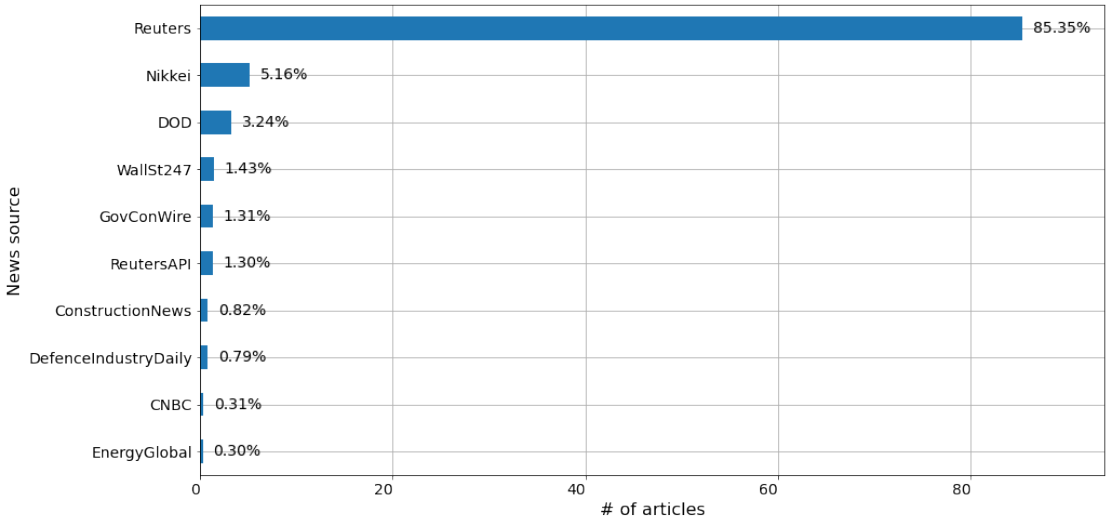


Figure 3.1. Amount of news articles per news source.

The original dataset is represented as a CSV file that has 13 columns. Table 3.1 shows the description of the columns. At the NER step, only a few of them are of interest, particularly *body* and *title*. However, the Named Entity recognition task is evaluated only on *body* column assuming that it contains similar or the same entities as the *title* would.
**Dataset of companies:** The companies dataset is collected by the partner company. It represents a CSV file containing information about 15396 public companies involved in public procurement globally. The structure of the dataset is described in Table 3.2. Unfortunately, the collected data can not be shared because it may have sensitive information that the partner company wouldn't like to disclose.

Table 3.1. Description of columns of the news articles dataset.

| Column name | Description |
|---|---|
| id | Internal article identifier |
| source | Article source identifier |
| category_url | The URL of source news category |
| url | URL of the article |
| language | 2-letter code for the original language (ISO 639-1) |
| title | Title of the article |
| body | Main text of the article |
| category | Category of the article |
| author | Author of the article |
| published_at_date | Original publication date |
| published_at_time | Original publication time |
| published_at_timezone | Original publication time timezone |
| last_retrieved_at | Time the article is retrieved from the source |

Table 3.2. Description of columns of the companies dataset.

| Column name | Description |
|---|---|
| id | Internal company identifier |
| name_1 | Full name of a company |
| name_2 | Optional variation of the company name |
| name_3 | Optional variation of the company name |
| ticker | Stock ticker of a publicly traded company |
| stock_exchange | Stock exchange name, where a company is publicly traded |

## 3.2 Data annotation

### 3.2.1 Annotation tool

There are different annotation tools available. Hence, comparison research is conducted to find the best match against such criteria as support of NE annotation, open-source code, simple deployment, the convenience of user experience (UX). In view of the fact that the article's data to be annotated might contain sensitive data, the distribution method preference falls into on-premise leaving Software-as-a-Service (SaaS) tools behind. Table 3.3 describes the most common open-source data annotation tools that support NE annotation as well as commercial services. The Pricing column includes such values as Open-source, Paid and Free. The Paid pricing plan means that the source code is proprietary and service fees apply; the Free plan means the source code is proprietary and no service fees apply. The Distribution column contains On-premise and Software-

as-a-Service (SaaS) values. The On-premise method means the consumer of a tool is responsible for the tool's deployment. The SaaS method means the tool is centrally hosted. The annotation tools are sorted by being free and open-source and by the amount of GitHub stars if applicable, which usually means the popularity of a project. Among the listed tools, the Label Studio annotation tool is selected. It supports NE annotation being open-source, user-friendly and supporting deployment on-premise.

Table 3.3. Data annotation tools that support NE annotation.

| **Name** | **Pricing** | **Distribution** | **GitHub stars** |
|---|---|---|---|
| Label Studio [63] | Open-source | On-premise | 7000 |
| Doccano [42] | Open-source | On-premise | 5500 |
| Universal Data Tool [58] | Open-source | On-premise, SaaS | 1500 |
| YEDDA [71] | Open-source | On-premise | 751 |
| INCEpTION [31] | Open-source | On-premise | 320 |
| Rubrix [57] | Open-source | On-premise | 188 |
| TALEN [59] | Open-source | On-premise | 106 |
| Kili Technology [61] | Free, Paid | SaaS | |
| Labelbox [32] | Free, Paid | SaaS | |
| TagTog [60] | Free, Paid | SaaS | |
| Amazon Mechanical Turk [26] | Paid | SaaS | |
| LightTag [46] | Paid | SaaS | |
| Prodigy [17] | Paid | SaaS | |
| UBIAI [53] | Paid | SaaS | |

### 3.2.2 Annotation of news articles with company identifiers

The news articles are annotated in a collaboration of the partner company and the thesis author efforts. They are annotated with internal company identifiers labelling company mentions in an article. As a result, a JavaScript Object Notation (JSON) file is produced containing news article identifiers, reference news article texts along internal identifiers of companies that are mentioned in related articles. Figure 3.2 displays a sample JSON file containing annotations of company mentions. Overall, 114 articles are annotated with 166 company IDs, where 144 companies are unique.

### 3.2.3 Annotation of news articles with company name entity

**Selection of articles:** Considering the number of news articles in the collected dataset, it is not practically possible to annotate the whole dataset in a reasonable time. Hence, the data annotation process is divided into two stages, where the first and second chunks

```
[
    {
        "article_id": "00000000",
        "source": "DOD",
        "url": "http://www.defense.gov/Newsroom/Contracts/Contract/Article/2023490/#14",
        "title": "Nov. 21, 2019 ARMY Contract #1",
        "body": "The Boeing Co., Mesa, Arizona, was awarded a $128,682,150 modification
(P00041) to Foreign Military Sales (Netherlands) W58RGZ-16-C-0023 for the Royal
Netherlands Air Force uniqueness on 11 Apache Attack Helicopter (AH)-64E aircraft,
recurring and non-recurring scope, version six integration, integrated logistics support,
product assurance, longbow crew trainers and initial peculiar ground support equipment.
Bids were solicited via the internet with one received. Work will be performed in Mesa,
Arizona, with an estimated completion date of June 30, 2025. Fiscal 2010 Foreign Military
Sales funds in the combined amount of $26,265,052 were obligated at the time of the
award. U.S. Army Contracting Command, Redstone Arsenal, Alabama, is the contracting
activity.",
        "annotations": [
            {
                "annotator_id": "bc85eb55",
                "result": {
                    "The Boeing Co.": "bf3e031c"
                }
            }
        ]
    }
]
```


Figure 3.2. Example of a news article annotated with company mentions.

of news are annotated, respectively. Taking into consideration the proportion of articles from Reuters source (85.35%), the distribution of articles selected for annotation drops the influence of Reuters source. The first annotation chunk is focused on such sources as Nikkei, Reuters and Department of Defence (DOD) primarily. Whereas, the second annotation chunk consists of articles from DOD and GovCon Wire sources only. The distribution of articles in the first, second and combined datasets is illustrated in Figure 3.3. The majority of articles belong to DOD and GovConWire globally, since these are the main sources of contract awards news, as expected. This is one of the requirements of business stakeholders of the partner company.

**Annotation guidelines:** To comply with the business requirements put on this project, the annotation guidelines are designed. They describe matching rules for the Company-Name entity type to be used by annotators:

- Annotate all types of companies — public and private;
- Annotate the longest match in singular form, including legal forms, such as "Tesla, Inc.", but not "Apple, Inc's";
- Annotate banks, law firms, investment companies, unless they refer to a specific entity but not to a general concept;
- Annotate news agencies, unless they are in a role of generic companies;

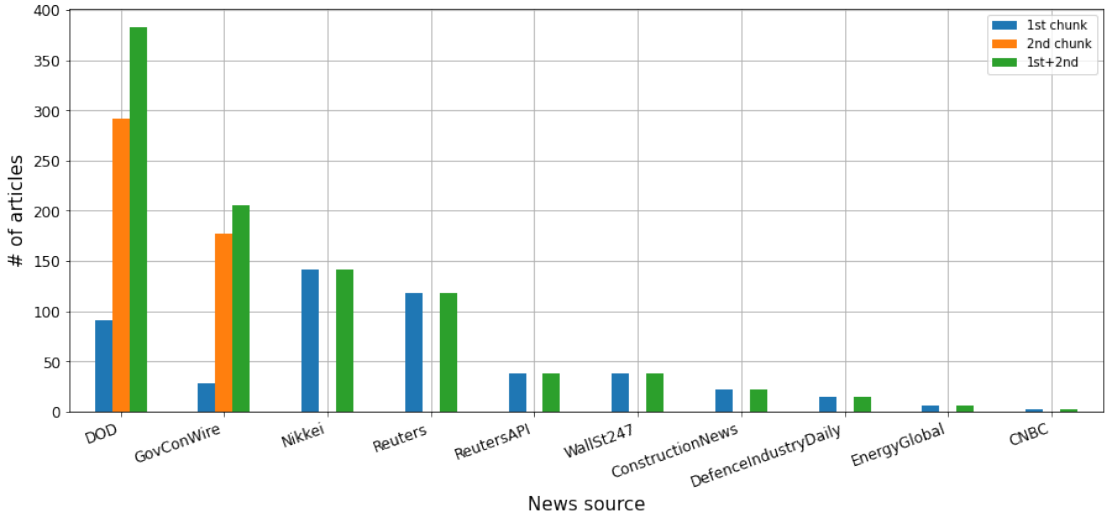


Figure 3.3. The distribution of articles in first, second chunks and the combined dataset.

- Do **not** annotate non-profit organizations as well as governmental organizations;
- Do **not** annotate abbreviated company names, such as the Bank of Japan mentioned as "BOJ";
- Do **not** annotate news agencies in their role as the news source, such as "...$800 million, Bloomberg reported";
- If in doubt of an entity being a company then search using generic sources, such as DuckDuckGo or Google search engines, or specialized sources, such as trade registers.

**Pre-annotation:** The major hypothesis that is formulated at this stage is that the use of pre-annotations would decrease the amount of time spent per news article during NE annotation. It has been shown that pre-annotation during the development of the first chunk for NER may increase the annotation speed [36]. The process of annotation is designed the following way:

1. Select a subset of articles chunk;
2. Annotate the subset manually and measure performance per article per hour;
3. Annotate the subset automatically, refine annotations manually and measure performance per article per hour;
4. Pick the most productive method and annotate the rest of the chunk.

The latest advancements in deep learning enable the application of language models to different tasks. One of interest is RoBERTa, a Robustly Optimized BERT Pretraining Approach. RoBERTa is a BERT-based Transformer language model that optimizes key hyper-parameters, training with much larger mini-batches and learning rates. It has been shown that RoBERTa is capable of outperforming other pre-training models on the NER task [67].

For the pre-annotation, a spaCy Python library is used [18]. It is commercial open-source software for NLP in Python, released under the MIT license. The spaCy library supports state-of-the-art solutions for NLP, including a pre-training pipeline with RoBERTa-base for NER, which is available under the "en_core_web_trf" tag in spaCy [13]. The pipeline was trained on blog texts, news and comments. The pipeline achieves 0.90 $F_1$-score, 0.90 precision and 0.90 recall on named entities, which include an ORG type, i.e. organization.

However, once the refinement of automatic annotations starts it appears that the automatic annotations contain a significant amount of false positives. The pre-training pipeline recognizes such entities as "World Health Organization" and "United Nations" as organizations. However, these entities cannot be described as companies, hence, such entities are false positives. Therefore, it is decided to proceed with manual annotation without the use of pre-annotation.

**Manual annotation:** Nevertheless, the original estimation of manual annotation effort was underrated. In reality, the performance of an annotator working with a 2000-character text on average with 1000 articles results in only 20 annotated articles per hour. If the numbers are projected, it would take 50 hours to annotate 1000 articles by a single rater. This way, the whole effort to annotate news articles is put by the internal powers of the partner company. For the sake of simplicity, in the early stages of work articles are annotated by one rater. However, this approach introduces a bias toward the rater. Being a human, a rater might not always label desired named entities as well as be prone to entity span mistakes. The data annotation is done by a 26-year old male, a resident of Bulgaria, currently pursuing a bachelor's degree in the field of Philosophy. He is an employee of the data and financial analysis department in the partner company.

**Output of manual data annotation:** The output of manual data annotation is a JSON file. The output file represents a list of objects, containing raw news article texts being annotated, internal article identifiers, Uniform Resource Locators (URL) of the news article source, the title of the business news source, article title and annotations. The annotations are presented in a form of a list of objects, each of them containing internal annotation ID, sub-string of the raw text being annotated as a Named Entity, NE type, start and end character positions of the entity. Figure 3.4 shows an example of data annotation output.

**Results of annotation:** Overall, 500 articles are annotated for the first chunk, 468 for the second chunk and 968 overall. There are 4527 entities annotated for the first chunk,

```
[
    {
        "article_id": "2239096",
        "source": "DOD",
        "url": "http://www.defense.gov/Newsroom/Contracts/Contract/Article/1525992/#10",
        "title": "May 18, 2018 NAVY Contract #5",
        "body": "Physical Optics Corp.,* Torrance, California, is awarded a $21,484,349 firm-fixed-
price modification (P00001) to a previously awarded contract (N00019-17-C-0078) for 102 F/A-18 E/F
and EA-18G data transfer units (DTUs) for the Navy and 50 DTUs for the government of Australia and
associated support equipment. Work will be performed in Torrance, California, and is expected to be
completed in August 2022. Fiscal 2017 aircraft procurement (Navy); and foreign military sales (FMS)
funds in the amount of $21,484,349 are being obligated at the time of award, none of which will
expire at the end of the current fiscal year. This contract combines purchases for the Navy
($15,280,606, 71 percent); and government of Australia ($6,203,743, 29 percent), under the FMS
program. The Naval Air Systems Command, Patuxent River, Maryland, is the contracting activity.",
        "annotations": [
            {
                "annotator_id": "bc85eb55",
                "result": [
                    {
                        "value": {
                            "start": 0,
                            "end": 21,
                            "text": "Physical Optics Corp.",
                            "labels": [
                                "CompanyName"
                            ]
                        },
                        "id": "cZbOaZv1IW",
                        "from_name": "labelBody",
                        "to_name": "article_body",
                        "type": "labels"
                    }
                ]
            }
        ]
    }
]
```


Figure 3.4. Example of NE annotation output.

2632 for the second and 7159 in total. Such news sources as Nikkei, GovConWire, Department of Defence (DOD) contain the most amount of CompanyName entities. The detailed distribution of amount of CompanyName entities per news source from two chunks combined is described in Figure 3.5.

**BILUO notation:** Such a data format is not suitable for the use with Machine Learning models, since it contains excessive information relevant for reference but not for training such models. Therefore, the JSON output is converted into Tab Separated Values (TSV) format, where data annotation labels are represented using BILUO notation. BILUO stands for Beginning, Inside and Last tokens of multi-token annotations, Unit-length annotations and Outside of annotation. BILUO notation clearly represents annotation boundaries as well as the token sequence. It is important that a NER model learns such sequences efficiently, since "Lockheed Martin Corporation" is not the same as "Martin Corporation Lockheed". BILUO notation represents a list of token-entity pairs separated by a tab character. Figure 3.6 shows a BILUO notation representation of such character sequence as "The Reserve Bank of India's Monetary Policy Committee on Thursday".

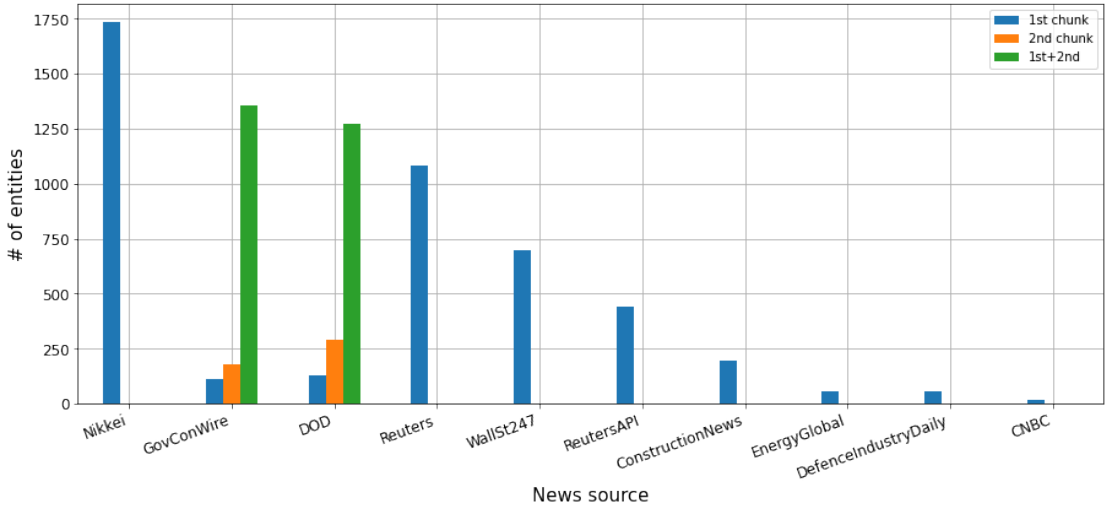


Figure 3.5. The distribution of entities in first, second chunks and the combined dataset.

**Data preprocessing:** Before the annotation output can be converted into BILUO notation, the raw texts have to be tokenized. Tokenization is a process of input character sequence separation. It is a crucial step in the sequence labelling task as it is required to understand, which sub-sequences are being classified. As a tokenizer, RoBERTa-base tokenizer was used, which is available under the "en_core_web_trf" tag in the spaCy library. However, the output of the tokenizer showed that there are 91 misaligned entities found during tokenization. This means that the tokenizer couldn't recognize token boundaries correctly. In case of the current data annotation output this happened due to the following reasons:

- Some words and sentences in news articles are concatenated, i.e. "end of sentence.Company";
- There are special characters concatenated with words, i.e. "@NAR.Takashimaya".

In addition, data annotation mistakes are found, which include:

- Some of the news sources are marked as companies, being a part of news article text, i.e. "Nikkei Asia" that is a news source;
- Some of the social media links are also marked as companies, where such news articles are reporting about other companies, i.e. "Twitter", "Linkedin";
- "'s" character sequence at the end of company names are marked as parts of CompanyName entity;

```
The     O
Reserve B-CompanyName
Bank    I-CompanyName
of      I-CompanyName
India   L-CompanyName
's      O
Monetary        O
Policy  O
Committee       O
on      O
Thursday        O
```

Figure 3.6. BILUO notation representation of character sequence.

- Other non-entity words are labelled as companies.

There are also tokenization errors found in the output:

- Some of "2%" string sequences appear as a single token, although have to be separate, i.e. "2 O\n% O";
- Some empty lines are tokenized and marked as O-type entities.

All the errors are eliminated manually and using scripting. Those entities that appear misaligned are marked with a "-" tag instead of an actual entity type. Such entities are labelled with BILUO tags once all the annotation and tokenization issues are resolved. Once the news article annotations are clean and represented in a suitable for ML models form they can be involved in Named Entity recognition system evaluation.

# 4 Methodology

In this chapter, the research methodology that this thesis follows is described. The source code produced by this work is available in a private repository [2]. The access rights have to be requested from the thesis authors. Figure 4.1 demonstrates the whole process that is divided into 5 modules, namely A (blue), B (yellow), C (green), D (purple) and E (pink):

- Module A (blue): the data is collected, processed and annotated. Business news articles are scrapped from several news sources. Along with that, a database of private and public companies is collected, in other words, a dataset of companies. Then news articles are annotated with company identifiers for the company detection task. Right after that, another set of news articles is annotated with the company name entity for the NER task, producing the first chunk of annotated articles. Chapter 3 reports about this module thoroughly;

- Module B (yellow): the direct lookup of company mentions is performed using Elasticsearch to partially answer the first research question. The chapter 5 describes the method in more detail;

- Module C (green): in this module the Name Entity recognition approach is presented. It starts with the evaluation of simple word-based models for NER on the first chunk of data. Later, the second chunk of NE annotations is produced, which is then combined with the first chunk. Subsequently, the simple models are evaluated on the combination of the first and second chunks of NE annotations. Afterwards, the Transformer-based models are evaluated on the combination of 1st and 2nd chunks. Directly after that, the best performing model is chosen and used to predict company name entities using the articles annotated with company identifiers. Please refer to sections 6-6.4 for evaluation details;

- Module D (purple): These predicted named entities are used to evaluate the application of NER for company detection task with two methods in this module. The first method records each company name of the companies dataset as a separate document in an Elasticsearch index and executes a search of indexed company names that uses the predicted company name entities as the search terms. The second approach is the opposite: it records predicted company name entities as documents in an Elasticsearch index and conducts a search of indexed Named Entities using company names from the dataset of companies as the search terms. The section 6.5 reports the evaluation results;

- Module E (pink): the three methods of the company detection task are compared in sub-process E, which is described in section 6.5.

[2] https://github.com/dalvsmerk/thesis2021

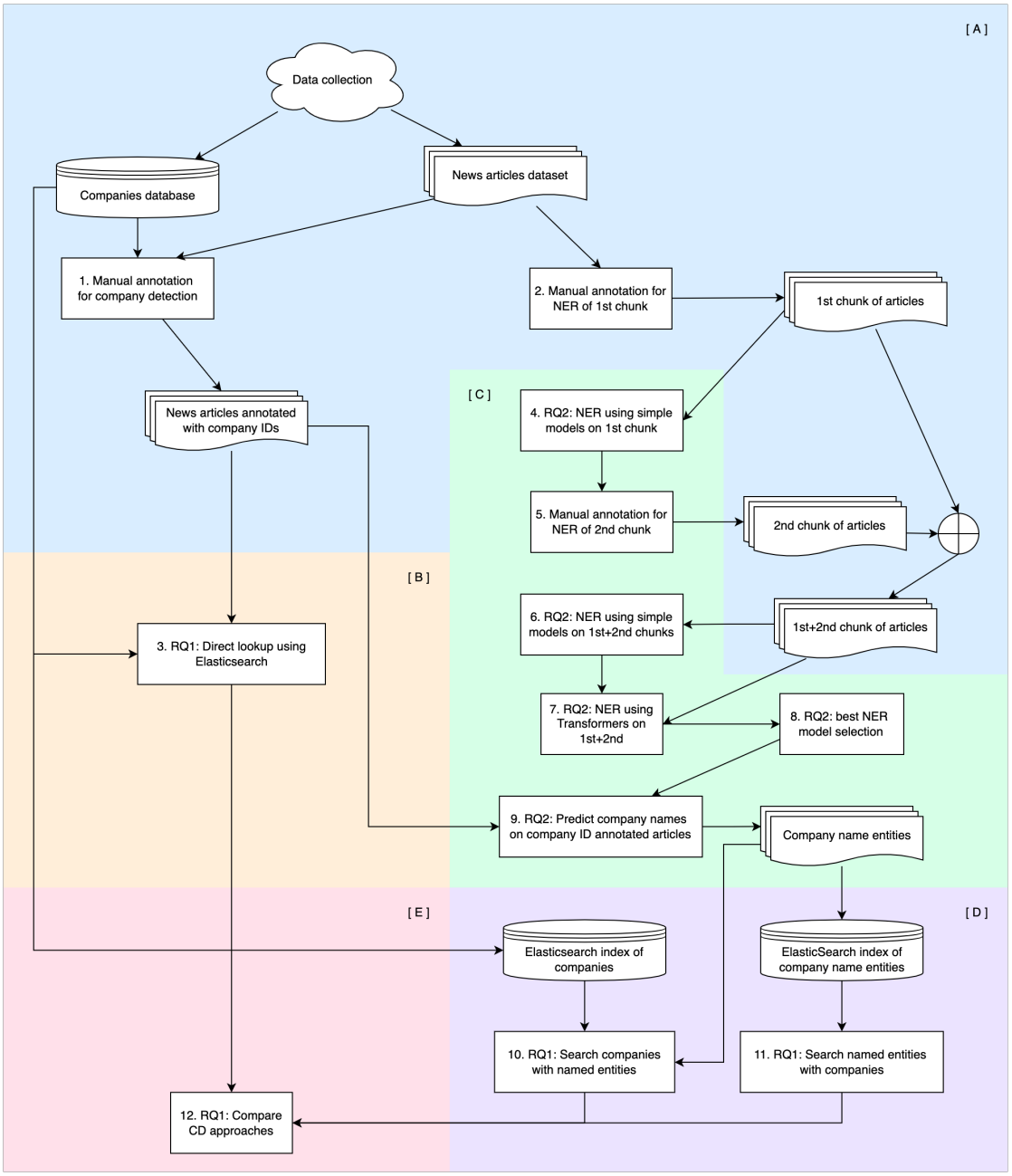


Figure 4.1. The process of thesis research.

# 5 RQ1: Naive approach to company detection

Since this thesis is focused on Named Entity recognition task, the company detection (CD) approach is not aimed to be advanced. The CD implementation serves as an evaluation platform for potential use of NER for CD. The so-called Naive approach involves the application of Elasticsearch search engine [24, 4]. In particular, to produce a comparison point, full-text search of companies on raw business news articles is implemented.

## 5.1 Method

Initially, the annotated news articles are indexed with Elasticsearch. Each articles is lemmatized, lower-cased and stop words are removed. Lemmatization is a process of word tranformation into its dictionary form [47]. Then a simple and straightforward algorithm proceeds: for each company in the dataset match its name with news articles using Match phrase query [5]. This query enables to adjust so-called *slop* parameter. It allows to control the deviation from the original phrase match. Given slop equal to 0 would match exactly on terms. In the experimental setup of this thesis the value of 2 is used: 2 terms of a query phrase are allowed to vary. For instance, given a query "General Motors Corporation", such phrases as "General Motors Co." or "Local Motors Corporation" would match. Since company names tend to be unique and short, it is expected that this approach would match company names precisely enough. Finally, the identifier of the matched company is appended to the set of predicted company IDs.

```
{
  "query": {
    "match_phrase": {
      "body": {
        "query": "Boeing Co.",
        "slop": 2
      }
    },

  },
  "highlight": {
    "fields": {
      "body": {

      }
    }
  }
}
```

Figure 5.1. Example query to match phrase with ElasticSearch.

The ElasticSearch search engine runs as a web-server over HTTP protocol, providing a RESTful API — Representational State Transfer Application Programming Interface. The experiments used the Python "elasticsearch" library [6] that serves as a client library, encapsulating HTTP requests with library methods. As an illustration of Elasticsearch

RESTful interface a sample query displayed on Figure 5.1 is executed as an HTTP request body in JSON format. Figure 5.2 presents an example response for such a query.


```
{
    "took": 2,
    "timed_out": false,
    "_shards": {
        "total": 1,
        "successful": 1,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": {
            "value": 3,
            "relation": "eq"
        },
        "max_score": 4.9860806,
        "hits": [
            {
                "_index": "news-articles-index",
                "_type": "_doc",
                "_id": "1783_0_6",
                "_score": 4.9860806,
                "_source": {
                    "article_id": "0000",
                    "body": "Boeing (NYSE: BA) received a potential two-year, $163.9M
amendment under a U.S. Navy contract for aircraft service life modification work. The company
said Friday it will continue to modernize the Navy's F/A-18 Super Hornet jets and establish a
second SLM line under the program..",
                    "entity_text": "Boeing",
                    "start_char": 0,
                    "end_char": 6,
                    "label": "CompanyName"
                },
                "highlight": {
                    "entity_text": [
                        "<em>Boeing</em>"
                    ]
                }
            }
        ]
    }
}
```


Figure 5.2. Example response of match phrase query with ElasticSearch.

## 5.2 Evaluation

In order to evaluate the approach performance such evaluation metrics as $F_1$-score (F), precision (P), recall (R) and Jaccard index (J) are exploited. *Jaccard index* is a statistic that measures similarity of two sets [28]. In case of this thesis, Jaccard index is used to measure similarity of the ground truth annotated set of company identifiers assigned to a news article and the set of predicted company IDs. Jaccard index is defined as follows, where A denotes the ground truth set of company ID and B denotes predicted set of company IDs:

$$J(A, B) = \frac{A \cap B}{A \cup B} \tag{1}$$

Table 5.1. Naive approach evaluation results.

| **Metrics set** | **Approach name** | **F** | **P** | **R** | **J** |
|---|---|---|---|---|---|
| Micro | Naive | 0.10 | 0.21 | 0.07 | 0.06 |
| Macro | Naive | 0.07 | 0.08 | 0.07 | 0.07 |

Two sets of metrics are computed over the evaluation results: macro and micro. The micro scores are defined as the sum of collected metrics over each company. The macro scores average metrics over each company. The results of Naive approach evaluation is presented in Table 5.1. It is visible that the method performs extremely poorly, barely matching company names in news articles. This thesis formulates a hypothesis that the application of Named Entity recognition can improve performance of the Naive approach.

# 6 RQ2: Company name entity recognition

The major assumption made at this step is that company detection requires company name recognition from news articles. In order to solve such a task some methods of Information Retrieval can be applied, such as Named Entity recognition.

The NER process starts from a simple approach, particularly simple NER models that were pre-trained on similar domain datasets. The simple models are evaluated on the first chunk of articles annotated with the CompanyName entity. Then, the same models are fine-tuned and evaluated on the first chunk. Next, another chunk is annotated and the simple models are fine-tuned and evaluated on the combination of first and second chunks. Finally, if the performance is still low enough experiments with Transformer-based NER models are conducted on the combination of both articles chunks. Figure 6.1 displays the whole NER process of this thesis. This chapter follows a similar structure to the described.

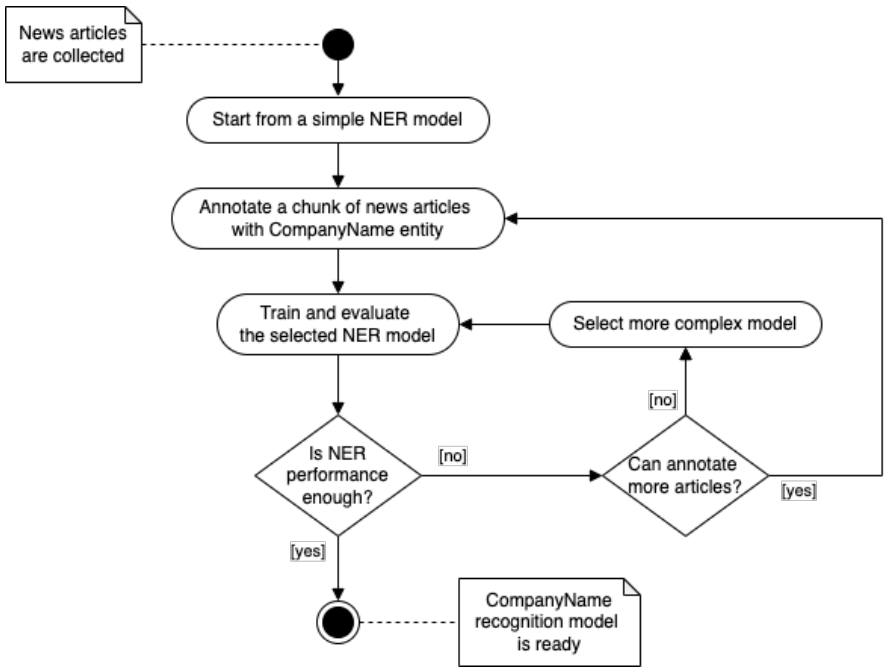


Figure 6.1. The activity diagram of Company Named Entity recognition.

## 6.1 NER evaluation metrics

From the side of potential real-world usage of company mention prediction, the cost of false positives is high: missing a company mention is better than identifying a wrong one. The high false positives rate would lead to ambiguous behaviour of company detection system that would confuse users and increase the Churn rate [50].

Although the only entity type being predicted is CompanyName, the amount of non-annotated tokens outweighs the amount of CompanyName entities in the annotated dataset. Therefore, it is not reasonable to use accuracy as an evaluation metric as the class distribution is highly unequal and false positives are considered highly costly. For this reason, such scores as $F$-score (F), precision (P) and recall (R) are involved. The precision and recall components are weighted with equal significance: the $\beta$ component of $F$-score is equal to 1. The $F_{\beta=1}$-score denoted as $F_1$-score and defined as

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{2}$$

where precision is defined as

$$precision = \frac{TP}{TP + FP} \tag{3}$$

and recall as

$$recall = \frac{TP}{TP + FN} \tag{4}$$

[43]. The $TP$, $FP$, $FN$ denote the following:

- Actual NEs that are predicted as NEs are called *true positives* or $TP$;
- Actual NEs that are **wronly** predicted as NEs are called *false positives* or $FP$;
- Actual NEs that are predicted as non-NEs are called *false negatives* or $FN$ [7].

## 6.2 The first NER phase: word-based

The initial set of NER experiments is aimed to confirm a hypothesis that company names can be predicted from business news articles using Named Entity recognition methods. Accordingly, Occam's razor is applied at this stage and simple models are evaluated, in particular word-based. Such NER models don't learn contextual dependencies between words and operate on character and/or word-level information.

### 6.2.1 Computation system setup

The computations involved in the first phase are done on a single CPU-powered node. Table 6.1 display the experimental setup for the first phase.

Table 6.1. First NER phase computation system setup.

| System characteristic | Value |
|---|---|
| Operating system (OS) | Ubuntu 20.04.2 LTS x86_64 |
| CPU | Intel i5-8265-U @ 3.8GHz |
| Disk Space | 512 GB |
| RAM | 16 GB |

### 6.2.2 Training and evaluation methodology

The amount of news articles involved in the first phase of the NER process is rather small. Consequently, such evaluation method as *single hold-out random subsampling* [3] would introduce a bias towards such a little training set. This method splits a dataset into train and hold-out subsets and trains a statistical model on the training subset and evaluates the hold-out subset. The first phase experiments apply *k-fold cross-validation* (CV) evaluation method to overcome the bias. Cross-validation is a data resampling method to assess the generalization ability of predictive models and to prevent overfitting [3]. Whereas k-fold cross-validation is a specific data resampling technique that splits a dataset into $k$ subsets and at each step trains a statistical model on $k - 1$ subsets (train split) and evaluates on the remaining subset (validation split). Figure 6.2 shows the process of model evaluation using 10-fold cross-validation.

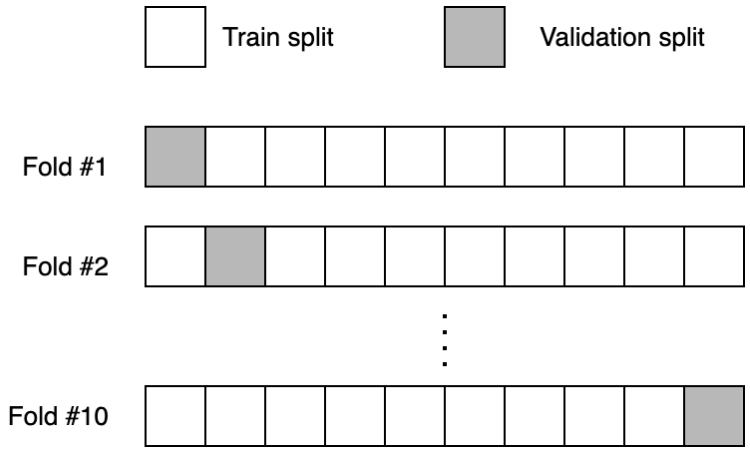


Figure 6.2. 10-fold cross-validation.

The 5-fold cross-validation is involved in the experiments using the first chunk of annotations. Consequently, at each fold, a model is trained on 400 articles and evaluated on 100. While the amount of annotated articles that is provided by the combination of first and second chunks allow the application of a single hold-out random subsampling

method for the rest of experiments. All the NER experiments apply the spaCy library for training and evaluation because it supports a wide range of NER models and can be easily integrated into the production environment as well as supports a simple and intuitive interface for NER training and evaluation.

### 6.2.3 Evaluation of pre-trained models on the first chunk of annotations

The spaCy library provides a set of English language modelling pipelines that support Named Entity recognition. In particular, the key interest of this stage is focused on pipelines available at such tags as "en_core_web_sm" (referred as Pre-trained WordNet), "en_core_web_md" (Pre-trained GloVe medium), "en_core_web_lg" (Pre-trained GloVe large) [14]. Each of them is pre-trained on written texts, such as news articles, blog posts, blog comments, which makes the domain area similar to the data domain of this thesis being business news. These pipelines were pre-trained on such entities as Person, Organization (ORG), Money, Time, etc. The models were not trained on the business news articles though.

Each of the pipelines consists of token-to-vector (tok2vec) and NER components. The token-to-vector component represents a model consisting of embedding layer and encoding layer [19]. The embedding layer learns the so-called "word vectors" that represent tokens in a common vector space. Then the encoding layer learns sentence matrix representation, where each row represents the meaning of each token in the context of the rest sentence [12]. Next, the sentence matrix is reduced to a single vector that can be fed to a fully connected (FC) layer, employing the mechanism of attention [44, 73]. Finally, the FC layer learns structured prediction serving as a controller for the transition-based parser in the NER component. The embedding layer benefits from the use of hash embedding that uses subword features, while the encoding layer consists of a Convolutional Neural network with a layer-normalized maxout activation function and residual connections [15]. The NER component implements a sequence tagging method, where the task of structure prediction is mapped to the state transition prediction. While the transition-based parser learns the state representation, the FC layer predicts scores for NE classes based on the state representation [16].

As the pre-trained vectors for the hash embedding layer GloVe medium and GloVe large vectors are used. Whereas the use of WordNet does not involve the application of pre-trained parameters. Global Vectors or GloVe is a model for unsupervised learning of word vectors [45]. It learns word representations in common vector space, so-called embeddings, based on word-word co-occurrence counts. One of the disadvantages of the model is that it works poorly with the words that are not included in the vocabulary. During training the pre-trained embedding vectors are static, i.e. are not updated. WordNet is a large lexical database of English. English nouns, verbs, adjectives and adverbs are organized into sets of synonyms, namely synsets, each representing a lexicalized concept [40]. The spaCy pipelines exploit synsets as a data augmentation technique.

Among all the supported labels only the ORG type is of interest. While CompanyName type refers only to for-profit organizations, ORG type refers to any type of organization, both for-profit and non-profit. Since CompanyName type refers to a subset of organizations it is assumed that the CompanyName entity type is similar to the ORG type at this stage. Therefore, the training and evaluation data annotations of the CompanyName entity were renamed to Organization type.

Table 6.2. Pre-trained spaCy NER pipelines evaluation results.

| **Data chunk** | **Model** | **F** | **P** | **R** |
|---|---|---|---|---|
| 1st | Pre-trained WordNet | 0.45 | 0.38 | 0.56 |
| | Pre-trained GloVe medium | 0.49 | 0.40 | 0.63 |
| | Pre-trained GloVe large | **0.52** | **0.42** | **0.68** |

The three experiments are conducted on the first chunk of annotated news articles. The results of the spaCy NER pipelines evaluation is displayed in Table 6.2. On business news articles the best performing pre-trained pipeline appears to be "en_core_web_lg" giving 0.52 $F_1$-score. Although, the gap between precision and recall is large — 0.26. This means that such a company name recognition system predicts a significant amount of false positives that can be explained by the ability of the models to predict non-profit organizations along with for-profit. Hence, non-profit organizations make up the majority of false positives. Nevertheless, these experiments confirm the original hypothesis of the NER methods' ability to predict company names from business news articles. This means that further experiments that involve model training should be conducted to improve the NER performance.

### 6.2.4 Fine-tuning of pre-trained models on the first chunk of annotations

Since the NER pipelines are already trained on news articles, blog posts and comments, it is possible to benefit from such a pre-training. In particular, instead of initializing model parameters with random values, parameters learned from the same domain tasks can be used for model parameters initialization. Whereas parameters of the new task are initialized randomly. This approach is called fine-tuning [35].

In current case, the token-to-vector component is initialized with parameters of "en_core_web_*" models, whereas the NER component is initialized randomly. Then the full model is trained on the annotated business news articles. The annotated news articles are fed into the pipeline, model updates the parameters using a mini-batch gradient descent algorithm [49]. At each epoch, the best performing model is saved to disk. Since the first annotation chunk is used for training and evaluation at this step, the 5-fold cross-validation method is used for evaluation. The fine-tuning process is described in Figure 6.3 [20]. The training hyper-parameters are displayed in Table 6.3.

Table 6.3. WordNet and GloVe fine-tuning training hyper-parameters.

| **Hyper-parameter** | **Value** |
|---|---|
| Optimizer | Adam |
| Adam $beta_1$ | 0.9 |
| Adam $beta_2$ | 0.999 |
| Adam $epsilon$ | 1e-9 |
| Learning rate | 0.001 |
| L2 regularization | 0.01 |
| Batch size | 32 |

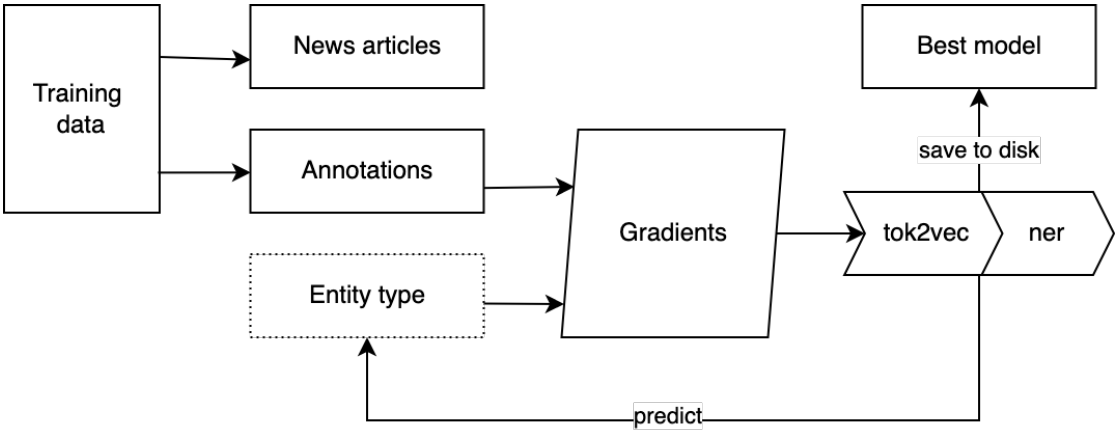


Figure 6.3. Fine-tuning of pre-trained WordNet and GloVe models for NER.

The fine-tuning approach resulted in higher $F_1$-score, precision and recall and more balanced precision and recall, which is visible in Table 6.4. While GloVe large model is much larger than the GloVe medium (43 MB vs 741 MB), the size of a model doesn't enable a model to predict CompanyName entities better. The medium model resulted in 0.75 $F_1$-score and 0.73 precision. At the same time, the large model resulted in 0.74 $F_1$-score and 0.72 precision with 0.77 recall of both models. The score difference is insignificant being only 0.01. Thus, at this point, it can be concluded that the increase of the amount of parameters wouldn't give better performance on the same amount of data with the same architecture. Therefore, more annotated news articles are required for NER performance improvement.

Table 6.4. Results of fine-tuning WordNet and GloVe models.

| Data chunk | Model | F | P | R |
|---|---|---|---|---|
| 1st | Fine-tuned WordNet | 0.58 | 0.69 | 0.5 |
| | Fine-tuned GloVe medium | **0.75** | **0.73** | **0.77** |
| | Fine-tuned GloVe large | 0.74 | 0.72 | **0.77** |

### 6.2.5 Fine-tuning of pre-trained models on combined chunks of annotations

From the results of previous experiments, it is visible that such models cannot produce satisfactory results with 500 annotated news articles. Consequently, at this stage of thesis, the second chunk of articles is annotated with the CompanyName entity. This decision follows the process described in Figure 6.1.

The same procedures are followed during fine-tuning of WordNet and GloVe models as well as training done with the same set of hyper-parameters. The difference is that this time the models are fine-tuned on a combined dataset of first and second chunks, i.e. on 968 news articles. The results of fine-tuning is presented in Table 6.5.

Table 6.5. Results of fine-tuning WordNet and GloVe models on the combined dataset.

| Data chunk | Model | F | P | R |
|---|---|---|---|---|
| 1st + 2nd | Fine-tuned WordNet | 0.65 | 0.63 | 0.68 |
| | Fine-tuned GloVe medium | **0.76** | **0.76** | **0.75** |
| | Fine-tuned GloVe large | 0.75 | 0.74 | 0.75 |

The use of more training data produced a slightly better performance compared to the training with the first chunk of data. Nevertheless, the NER model performance is still low to be used in real-world products. Therefore, it is required to apply more sophisticated model architectures.

## 6.3 Second NER phase: Transformer-based

Transformers can effectively learn the significance of words based on context. Therefore, it can be assumed that such a context-based approach would improve the NER performance significantly compared to the word-based approaches such as GloVe. The neural models usually contain three components: word embedding layer, context encoder layer and decoder layer [69]. In the context of the experiments of this thesis Transformer models represent word embedding and context encoder layers. While the decoder layer is represented by the Transition-Based Chunking Model [33] in a combination with a Fully-Connected layer that predicts named entities from the state representation that the transition-based algorithm learns. This thesis employs the RoBERTa-base model as the

Transformer layer [37]. All the RoBERTa-based experiments, except the pre-training model evaluation, are performed with the same hyper-parameters that are presented in Table 6.6 based on recommended settings from spaCy configuration. The learning rate is set to a low value in order to preserve some of the representational structure learned in the original RoBERTa tasks [35].

Table 6.6. RoBERTa-base fine-tuning training hyper-parameters.

| **Hyper-parameter** | **Value** |
|---|---|
| Optimizer | Adam |
| Adam $beta_1$ | 0.9 |
| Adam $beta_2$ | 0.999 |
| Adam $epsilon$ | 1e-9 |
| Dropout rate | 0.1 |
| Learning rate | 5e-5 |
| L2 regularization | 0.01 |
| Batch size | 128 |

### 6.3.1 Computation system setup

The computations involved in the second phase are done on a single GPU-powered node. Table 6.7 displays the experimental setup for the second NER phase. The choice of GPU-powered system use is based on its better parallel computing capabilities when training artificial neural networks. The experiments conducted with CPU-based and GPU-based systems show that the use of GPU-based systems reduces training time by 10 times compared to the CPU-based system on the same neural architecture.

Table 6.7. The second NER phase computation system setup.

| **System characteristic** | **Value** |
|---|---|
| Operating system (OS) | CentOS 7 x86_64 |
| CPU | 2 x Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz |
| GPU | 2 x NVIDIA Tesla V100 16 GB VRAM |
| Disk Space | 2.1 TB |
| RAM | 512 GB |

### 6.3.2 Evaluation of pre-trained Transformer from spaCy parameters

The pre-trained model initialized with "en_core_web_trf" parameters from spaCy [13] is evaluated on the combination of first and second chunks of data. At this stage, it is assumed that the Organization entity type is similar to the CompanyName entity type. It can be expected to have a significant amount of false positives. However, this approach serves as a comparison point for further experiments. The evaluation results are available in Table 6.8.

Since recall score is the highest and precision score is low it is visible that such a model can predict much more entities than previous models. However, many of them are false positives. Considering that the CompanyName entity type is a subset of the Organization entity type, RoBERTa-base is already learned on a similar domain. Therefore, it makes sense to fine-tune this model on business news articles.

Table 6.8. Results of RoBERTa-base Transformer model from spaCy parameters evaluation.

| **Data chunk** | **Model** | **F** | **P** | **R** |
|---|---|---|---|---|
| 1st + 2nd | RoBERTa-base from spaCy | 0.66 | 0.52 | 0.89 |

### 6.3.3 Fine-tuning pre-trained RoBERTa-base

The next step is to try to train RoBERTa-base [37] from its default pre-training parameters with a NER component initialized randomly on top. The original RoBERTa-base is pre-trained on English Wikipedia, English news articles and web content. Thus, its primary domain is similar to the data domain of this thesis. The model is trained on the combination of first and second chunks of annotated articles. This approach updated parameters of both Transformer and NER layers during training.

Table 6.9. Results of fine-tuning pre-trained default RoBERTa-base.

| **Data chunk** | **Model** | **F** | **P** | **R** |
|---|---|---|---|---|
| 1st + 2nd | Fine-tuned default RoBERTa-base | 0.85 | 0.85 | 0.85 |

The result of fine-tuning RoBERTa-base initialized from default parameters is presented in Table 6.9. It is visible that the performance gain is significant with this approach, compared to the pre-trained RoBERTa-base evaluation and fine-tuning of word-based models. While fine-tuned GloVe medium resulted in 0.76 $F_1$-score, default RoBERTa-base outputs 0.85 $F_1$-score, with 0.1 score improvement. In addition, such an approach is able to predict many company names and the bulk of them are relevant — the result of balanced precision and recall.

### 6.3.4 Fine-tuning pre-trained RoBERTa-base with frozen Transformer

One of the approaches for fine-tuning pre-trained models is to freeze some of the high-level layers of a neural network [35]. This means that the parameters of such layers wouldn't be updated during training. This approach enables the reduction of model overfitting. At the same time, the NER layer is initialized randomly. The training results are visible in Table 6.10.

Table 6.10. Results of fine-tuning pre-trained default RoBERTa-base with frozen Transformer.

| **Data chunk** | **Model** | **F** | **P** | **R** |
|---|---|---|---|---|
| 1st + 2nd | Fine-tuned default RoBERTa-base frozen | 0.70 | 0.69 | 0.72 |

Since the Transformer freezing approach doesn't help to raise performance, it can be concluded that the pre-trained default RoBERTa-base model lacks knowledge regarding a particular domain of business news. At the same time, the NER layer can successfully predict relevant entities out of the amount of total predicted — precision and recall are more or less balanced.

### 6.3.5 Fine-tuning pre-trained RoBERTa-base from spaCy parameters

Another approach to be evaluated is the use of pre-trained parameters from the spaCy library, in particular the "en_core_web_trf" model [13]. This model is trained with the English subset of the OntoNotes 5.0 dataset, which is an annotated collection of various genres of text, including news, conversational telephone speech, weblogs, Usenet newsgroups, broadcast, talk shows [68]. The pre-training domain of the model is similar to the business news domain, thus it is expected to achieve valuable results. During training, both Transformer and NER layer parameters are updated. Whereas, the Transformer parameters are initialized with spaCy "en_core_web_trf" weights and the NER parameters are initialized randomly. The output model performance is presented in Table 6.11. As a result, the trained system reaches $F_1$-score of 0.86, while precision and recall are well balanced — 0.86 and 0.87, respectively.

Table 6.11. Results of fine-tuning pre-trained spaCy RoBERTa-base.

| **Data chunk** | **Model** | **F** | **P** | **R** |
|---|---|---|---|---|
| 1st + 2nd | Fine-tuned spaCy RoBERTa-base | 0.86 | 0.86 | 0.87 |

### 6.3.6 Fine-tuning pre-trained RoBERTa-base from spaCy parameters with frozen Transformer

Finally, a pre-trained RoBERTa-base is trained, initialized with spaCy parameters. At this stage, the Transformer layer is frozen, i.e. it's parameters are not updated during training. The NER parameters are initialized randomly. The end result of training is displayed in Table 6.12. In consequence, the model achieved only 0.79 $F_1$-score, 0.76 precision and 0.82 recall. This approach enables recognition of many named entities but the amount of actual company names is low.

Table 6.12. Results of fine-tuning pre-trained spaCy RoBERTa-base with frozen Transformer.

| **Data chunk** | **Model** | **F** | **P** | **R** |
|---|---|---|---|---|
| 1st + 2nd | Fine-tuned spaCy RoBERTa-base frozen | 0.79 | 0.76 | 0.82 |

## 6.4 CompanyName entity recognition results

According to the outcomes of the experiments, the fine-tuned spaCy RoBERTa-base performs the best, achieving 0.86 $F_1$-score, 0.86 precision and 0.87 recall. As a result of well-balanced precision and recall the model predicts a high-enough quantity of named entities, the bulk of which are actually company names. This model outperforms fine-tuned default RoBERTa-base just by a 0.01-0.02 score value. Table 6.13 summarizes evaluation results of different NER methods applied in this thesis.

Although the Transformer-based approach produces the best results compared to the other experiments this comes with higher computation power requirements. While GloVe models can efficiently be trained using CPUs, Transformer-based models require GPU power as well as a larger amount of disk space and Random Access Memory (RAM). Therefore, a trade-off between model performance and required computation power appears. Since the NE recognition step is required to run neither in real-time nor in near real-time, more computation power should be allocated for better performance.

Considering the relatively high performance of the resulting NER system, the fine-tuned spaCy RoBERTa-base is used to predict company name entities, which can be involved in company detention now. Although, the fine-tuned default RoBERTa-base model would produce similar results.

Table 6.13. NER evaluation results.

| Data chunk | Model | F | P | R |
|---|---|---|---|---|
| 1st | Pre-trained WordNet | 0.45 | 0.38 | 0.56 |
| | Pre-trained GloVe medium | 0.49 | 0.40 | 0.63 |
| | Pre-trained GloVe large | 0.52 | 0.42 | 0.68 |
| | Fine-tuned WordNet | 0.58 | 0.69 | 0.50 |
| | Fine-tuned GloVe medium | 0.75 | 0.73 | 0.77 |
| | Fine-tuned GloVe large | 0.74 | 0.72 | 0.77 |
| 1st + 2nd | Fine-tuned WordNet | 0.65 | 0.63 | 0.68 |
| | Fine-tuned GloVe medium | 0.76 | 0.76 | 0.75 |
| | Fine-tuned GloVe large | 0.75 | 0.74 | 0.75 |
| | Pre-trained RoBERTa-base | 0.66 | 0.52 | 0.89 |
| | Fine-tuned default RoBERTa-base | 0.85 | 0.85 | 0.85 |
| | Fine-tuned default RoBERTa-base frozen | 0.70 | 0.69 | 0.72 |
| | Fine-tuned spaCy RoBERTa-base | **0.86** | **0.86** | **0.87** |
| | Fine-tuned spaCy RoBERTa-base frozen | 0.79 | 0.76 | 0.82 |

## 6.5 Application of NER on company detection

This thesis applies Named Entity recognition as a pre-processing step for the company detection task. The CompanyName entity is predicted on the business news articles. Then, two methods that extend the Naive approach are evaluated. First, the CompanyName entity is predicted on the news articles annotated with company identifiers. Then, the first method does a Match phrase search using company names from the companies dataset within the predicted Named Entities. Finally, the second method accomplishes Match phrase search using predicted named entities within available company names extracted from the company dataset.

**Search mentions of companies in predicted named entities:** Initially, the predicted named entities are indexed with ElasticSearch and then search of companies from the companies dataset is conducted over the indexed named entities using Match phrase query of ElasticSearch.

**Search mentions of predicted named entities in companies:** The second approach is the opposite of the first one. The company dataset is indexed with ElasticSearch. Next, each predicted named entity is searched using the Match phrase query over the indexed companies.

**Summary of experiments:** The search results are evaluated against $F_1$-score, precision, recall and Jaccard index. Table 6.14 presents the total results of three experiments conducted to solve the company detection task. Obviously, the selected approach for company detection performs extremely poorly on given datasets. However, the introduction of Named Entity recognition significantly improves the results. The effect of

Table 6.14. Naive approach evaluation results.

| **Metrics set** | **Approach name** | **F** | **P** | **R** | **J** |
|---|---|---|---|---|---|
| Micro | Naive | 0.10 | 0.21 | 0.07 | 0.06 |
| | Companies within Named Entities | 0.09 | 0.67 | 0.05 | 0.05 |
| | Named Entities within companies | **0.28** | **0.81** | **0.17** | **0.17** |
| Macro | Naive | 0.07 | 0.08 | 0.07 | 0.07 |
| | Companies within Named Entities | 0.05 | 0.06 | 0.04 | 0.04 |
| | Named Entities within companies | **0.16** | **0.17** | **0.16** | **0.16** |

high precision and low recall is visible: the second approach is capable of predicting an insignificant amount of companies. At the same time, the bulk of predicted companies are relevant, i.e. predicted company IDs match with the annotations greatly.

# 7 Conclusions

This thesis introduces a Transformer-based approach for company name recognition from business news articles. It is shown that the application of context-based NER models can improve the performance significantly at a cost of increased computation power requirements compared to the use of non-context-based NER models. In addition, company mention detection can benefit from the use of NER as a pre-processing step of news articles.

The application of Transformer-based NER provides a major improvement of CD in terms of the diversity of company names recognized as well as the relevance of the detected companies. Furthermore, such an approach resolves a task of open-world company mention detection. Some real-world products may benefit from recognizing unknown companies augmenting the existing company database.

While the presented approach results in a well-performing model, there is still space for improvement. First of all, some models can benefit from the exploitation of such contextual information like company name variations [39]. Then, more news articles can be annotated containing more heterogeneous company names. Next, more data raters should be employed in the process of data annotation. This thesis demonstrates that annotation errors might occur during the annotation procedure, including wrong entity labelling and incorrect entity boundaries. Besides, NER annotations tend to appear biased towards the annotator so an inter-annotator agreement has to be concluded, i.e. using Cohen's kappa statistic [9]. Finally, more Named Entities can be recognized that relate to companies, such as stock ticker, address, area of expertise, etc.

# References


[1] Sherief Abdallah, Khaled Shaalan, and Muhammad Shoaib. Integrating rule-based system with classification for arabic named entity recognition. In International Conference on Intelligent Text Processing and Computational Linguistics, pages 311–322. Springer, 2012.

[2] Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. Tuning multilingual transformers for named entity recognition on slavic languages. BSNLP'2019, page 89, 2019.

[3] Daniel Berrar. Cross-validation., 2019.

[4] Elasticsearch B.V. elastic/elasticsearch: Free and Open, Distributed, RESTful Search Engine. https://github.com/elastic/elasticsearch.

[5] Elasticsearch B.V. Match phrase query | Elasticsearch Guide [7.16] | Elastic. https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query-phrase.html.

[6] Elasticsearch B.V. Python Elasticsearch Client — Elasticsearch 7.16.2 documentation. https://elasticsearch-py.readthedocs.io/en/v7.16.2/.

[7] Davide Chicco and Giuseppe Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. BMC genomics, 21(1):1–13, 2020.

[8] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. Transactions of the Association for Computational Linguistics, 4:357–370, 2016.

[9] Jacob Cohen. A coefficient of agreement for nominal scales. Educational and psychological measurement, 20(1):37–46, 1960.

[10] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. Journal of machine learning research, 12(ARTICLE):2493–2537, 2011.

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.

[12] Explosion. Embed, encode, attend, predict: The new deep learning formula for state-of-the-art NLP models · Explosion. https://explosion.ai/blog/deep-learning-formula-nlp.

[13] Explosion. English · spaCy Models Documentation. https://spacy.io/models/en#en_core_web_trf.

[14] Explosion. English · spaCy Models Documentation. https://spacy.io/models/en.

[15] Explosion. Model Architectures · spaCy API Documentation. https://spacy.io/api/architectures#HashEmbedCNN.

[16] Explosion. Parsing English in 500 Lines of Python · Explosion. https://explosion.ai/blog/parsing-english-in-python.

[17] Explosion. Prodigy · An annotation tool for AI, Machine Learning NLP. https://prodi.gy/.

[18] Explosion. spaCy · Industrial-strength Natural Language Processing in Python. https://spacy.io/.

[19] Explosion. Tok2Vec · spaCy API Documentation. https://spacy.io/api/tok2vec.

[20] Explosion. Training Pipelines Models · spaCy Usage Documentation. https://spacy.io/usage/training.

[21] Dimitra Farmakiotou, Vangelis Karkaletsis, John Koutsias, George Sigletos, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. Rule-based named entity recognition for greek financial texts. In Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000), pages 75–78. Citeseer, 2000.

[22] Eduardo Ferreira, João Balsa, and António Branco. Combining rule-based and statistical methods for named entity recognition in portuguese. In Actas da 5a Workshop em Tecnologias da Informaçao e da Linguagem Humana, 2007.

[23] Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. Named entity recognition through classifier combination. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, pages 168–171, 2003.

[24] Clinton Gormley and Zachary Tong. Elasticsearch: the definitive guide: a distributed real-time search and analytics engine. " O'Reilly Media, Inc.", 2015.

[25] Ralph Grishman and Beth M Sundheim. Message understanding conference-6: A brief history. In COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics, 1996.

[26] Amazon.com Inc. Amazon Mechanical Turk, howpublished = https://www.mturk.com/, timestamp = 2021.12.11.

[27] Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In COLING 2002: The 19th International Conference on Computational Linguistics, 2002.

[28] Paul Jaccard. The distribution of the flora in the alpine zone. 1. New phytologist, 11(2):37–50, 1912.

[29] Jisha P Jayan, RR Rajeev, and Elizabeth Sherly. A hybrid statistical approach for named entity recognition for malayalam language. In Proceedings of the 11th Workshop on Asian Language Resources, pages 58–63, 2013.

[30] Muhammad Raza Khan, Morteza Ziyadi, and Mohamed AbdelHady. Mt-bioner: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers. arXiv preprint arXiv:2001.08904, 2020.

[31] Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations, pages 5–9, Santa Fe, New Mexico, 2018.

[32] Labelbox. Labelbox/labelbox: Labelbox is the fastest way to annotate data to build and ship computer vision applications. https://github.com/Labelbox/labelbox.

[33] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360, 2016.

[34] Andre Lamurias and Francisco M Couto. Lasigebiotm at mediqa 2019: Biomedical question answering using bidirectional transformers and named entity recognition. In Proceedings of the 18th BioNLP workshop and shared task, pages 523–527, 2019.

[35] Zhizhong Li and Derek Hoiem. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12):2935–2947, 2017.

[36] Todd Lingren, Louise Deleger, Katalin Molnar, Haijun Zhai, Jareen Meinzen-Derr, Megan Kaiser, Laura Stoutenborough, Qi Li, and Imre Solti. Evaluating the impact of pre-annotation on annotation speed and potential bias: natural language processing gold standard development for clinical named entity recognition in clinical

trial announcements. Journal of the American Medical Informatics Association, 21(3):406–413, 2014.

[37] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[38] Alireza Mansouri, Lilly Suriani Affendey, and Ali Mamat. Named entity recognition approaches. International Journal of Computer Science and Network Security, 8(2):339–344, 2008.

[39] Andrei Mikheev, Marc Moens, and Claire Grover. Named entity recognition without gazetteers. In Ninth Conference of the European Chapter of the Association for Computational Linguistics, pages 1–8, 1999.

[40] George A Miller. Wordnet: a lexical database for english. Communications of the ACM, 38(11):39–41, 1995.

[41] Behrang Mohit. Named entity recognition. In Natural language processing of semitic languages, pages 221–245. Springer, 2014.

[42] Hiroki Nakayama, Takahiro Kubo, Junya Kamura, Yasufumi Taniguchi, and Xu Liang. doccano: Text annotation tool for human, 2018. Software available from https://github.com/doccano/doccano.

[43] David L Olson and Dursun Delen. Advanced data mining techniques. Springer Science & Business Media, 2008.

[44] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933, 2016.

[45] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014.

[46] Tal Perry. Lighttag: Text annotation platform. arXiv preprint arXiv:2109.02320, 2021.

[47] Martin F Porter. An algorithm for suffix stripping. Program, 1980.

[48] Ahmed Cheikh Rouhou, Marwa Dhiaf, Yousri Kessentini, and Sinda Ben Salem. Transformer-based approach for joint handwriting and named entity recognition in historical document. Pattern Recognition Letters, 2021.

[49] Sebastian Ruder. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747, 2016.

[50] Salesforce.com. How to Calculate Customer Churn Rate and Revenue Churn Rate - Salesforce.com. https://www.salesforce.com/resources/articles/how-calculate-customer-churn-and-revenue-churn/.

[51] B Sasidhar, PM Yohan, A Vinaya Babu, and A Govardhan. Named entity recognition in telugu language using language dependent features and rule based approach. International Journal of Computer Applications, 22(8):30–34, 2011.

[52] Stefan Schweter and Alan Akbik. Flert: Document-level features for named entity recognition. arXiv preprint arXiv:2011.06993, 2020.

[53] UBIAI Web Services. UBIAI Easy to Use Text Annotation Tool | Upload documents, start annotating, and create advanced NLP model in a few hours. https://ubiai.tools/.

[54] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In Proceedings of the international joint workshop on natural language processing in biomedicine and its applications (NLPBA/BioNLP), pages 107–110, 2004.

[55] Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. arXiv preprint arXiv:1707.05928, 2017.

[56] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena, 404:132306, 2020.

[57] Open source community of GitHub. recognai/rubrix: Python framework to explore, label, and monitor data for NLP projects. https://github.com/recognai/rubrix.

[58] Open source community of GitHub. UniversalDataTool/universal-data-tool: Collaborate label any type of data, images, text, or documents, in an easy web interface or desktop app. https://github.com/UniversalDataTool/universal-data-tool.

[59] Dan Roth Stephen Mayhew. Talen: Tool for annotation of low-resource entities. In ACL System Demonstrations, 2018.

[60] tagtog. tagtog · AI-enabled Text Annotation Tool | PDF, Markdown, CSV, html, tweets, many more Document types. https://www.tagtog.net/.

[61] Kili Technology. kili-technology/kili-playground: Simplest and fastest image and text annotation tool. https://github.com/kili-technology/kili-playground.

[62] Shubo Tian, Arslan Erdengasileng, Xi Yang, Yi Guo, Yonghui Wu, Jinfeng Zhang, Jiang Bian, and Zhe He. Transformer-based named entity recognition for parsing clinical trial eligibility criteria. In Proceedings of the 12th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, pages 1–6, 2021.

[63] Maxim Tkachenko, Mikhail Malyuk, Nikita Shevchenko, Andrey Holmanyuk, and Nikolai Liubimov. Label Studio: Data labeling software, 2020-2021. Open source software available from https://github.com/heartexlabs/label-studio.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[65] Hanna M Wallach. Conditional random fields: An introduction. Technical Reports (CIS), page 22, 2004.

[66] Chuan Wang, Nianwen Xue, and Sameer Pradhan. A transition-based algorithm for amr parsing. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 366–375, 2015.

[67] Yu Wang, Yining Sun, Zuchang Ma, Lisheng Gao, Yang Xu, and Ting Sun. Application of pre-training models in named entity recognition. In 2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), volume 1, pages 23–26. IEEE, 2020.

[68] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. Ontonotes release 5.0 ldc2013t19. Linguistic Data Consortium, Philadelphia, PA, 23, 2013.

[69] Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. Tener: adapting transformer encoder for named entity recognition. arXiv preprint arXiv:1911.04474, 2019.

[70] Rongen Yan, Xue Jiang, and Depeng Dang. Named entity recognition by using xlnet-bilstm-crf. Neural Processing Letters, pages 1–18, 2021.

[71] Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. Yedda: A lightweight collaborative text span annotation tool. 2018.

[72] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32, 2019.

[73] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pages 1480–1489, 2016.

# Appendix

# I. Glossary

| | |
|---|---|
| API | Application Programming Interface |
| BERT | Bidirectional Encoding Representations from Transformers |
| BiLSTM | Bidirectional Long-Short Term Memory |
| CD | Company detection |
| CNN | Convolutional Neural Network |
| CPU | Central Processing Unit |
| CRF | Conditional Random Field |
| CSV | Comma-separated values |
| CV | Cross-validation |
| DL | Deep Learning |
| FC | Fully-Connected |
| GPU | Graphics Processing Unit |
| GloVe | Global Vectors |
| HTTP | Hypertext Transfer Protocol |
| ID | Identifier |
| IEHHR | Information Extraction in Historical Handwritten Records |
| JSON | JavaScript Object Notation |
| LSTM | Long-Short Term Memory |
| ML | Machine Learning |
| NE | Named Entity |
| NER | Named Entity recognition |
| NLI | Natural Language Inference |
| NLP | Natural Language Processing |
| QA | Question Answering |
| RAM | Random Access Memory |
| REST | Representational State Transfer |
| RNN | Recurrent Neural Network |
| RQE | Recognizing Question Entailment |
| RoBERTa | Robustly Optimized BERT Pretraining Approach |
| SVM | Support Vector Machine |
| TSV | Tab-separated values |

# II. Licence

## Non-exclusive licence to reproduce thesis and make thesis public

I, **Vladyslav Umerenko**,
(author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   **Transformer-based Approach for Named Entity Recognition from Business News Articles for Company Detection**,
   (title of thesis)

   supervised by Rajesh Sharma.
   (supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Vladyslav Umerenko
***06/01/2022***