Dissecting Company Names using Sequence Labeling

Michael Loster[1], Manuel Hegner[1], Felix Naumann[1], and Ulf Leser[2]

[1] Hasso Plattner Institute, University of Potsdam, Germany
{michael.loster, manuel.hegner, felix.naumann}@hpi.de
[2] Department of Computer Science, Humboldt-Universität zu Berlin, Germany
leser@informatik.hu-berlin.de

Abstract. Understanding the inherent structure of company names by identifying their constituent parts yields valuable insights that can be leveraged by other tasks, such as named entity recognition, data cleansing, or deduplication. Unfortunately, segmenting company names poses a hard problem due to their high structural heterogeneity. Besides obvious elements, such as the core name or legal form, company names often contain additional elements, such as personal and location names, abbreviations, and other unexpected elements.

While others have addressed the segmentation of person names, we are the first to address the segmentation of the more complex company names. We present a solution to the problem of automatically labeling the constituent name parts and their semantic role within German company names. To this end we propose and evaluate a collection of novel features used with a conditional random field classifier. In identifying the constituent parts of company names we achieve an accuracy of 84%, while classifying the colloquial names resulted in an F1 measure of 88%.

1 Introduction

Telling apart the different components of company names can be of high value for many applications. Therefore, we aim to recognize and automatically label the different constituent parts of company names using a predefined set of labels. Following [13], we formalize this problem as a supervised sequence labelling task. Given a sequence of tokens $t_1, \ldots, t_n$, for each token t_k we aim to predict the most probable label $l_i \in T, 1 \leq i \leq |T|$ from a specifically designed set of labels T, which we introduce in Section 3.

This problem is difficult to solve due to the extreme variety of company names and in turn, their constituent parts. In addition to their name and legal form, they often contain other elements, such as person names, abbreviations or industry-specific information. Consider company names such as “Dr. Ing. h.c. F. Porsche AG” or “ABB AVUS Bücherdienst Berlin Verlags- und Vertriebs GmbH”. Alongside information about a person title (“Dr. Ing. h.c.”), the location (“Berlin”), or the industry sector (“Verlags- und Vertriebs”) they also contain abbreviations, such as “F.”, which are especially hard to resolve. Here

"F." abbreviates "Ferdinand", the founder of the company Porsche. This structural diversity is problematic for many subsequent tasks, as it often negatively affects their results.

Many applications, such as named entity recognition (NER), entity resolution, or duplicate detection, benefit from knowing the inherent structure of company names. By systematically decomposing them into their constituent parts, it is possible to leverage the semantic meaning of each part to improve the performance of the aforementioned systems. Consider, for example, a NER system trained to detect company names in textual data. As we have shown, the performance of a machine learning-based NER system can be significantly improved by incorporating additional domain knowledge into the training process [7]. Using official and publicly accessible sources for the compilation of domain knowledge leads to dictionaries, which usually consist of formal company names, such as "Dr. Ing. h.c. F. Porsche AG". Unfortunately, textual documents, such as newspaper articles, often refer to companies by their colloquial names, i.e., "Porsche", which is generally shorter and more convenient than using their full legal name. Even worse, most companies have multiple valid colloquial names, hereafter referred to as alias names by which they are mentioned. Considering again Porsche, the company has several alias names, namely "Ferdinand Porsche AG", "Porsche AG", or just plain "Porsche". During the processing of such names, the semantic knowledge derived from its constituent parts can be used to automatically generate possible alias names, such as those mentioned above. In revealing the inherent structure of company names by focusing on the classification of their constituent parts we develop a deeper understanding of how company names are constructed. Moreover, the name alone can be used to derive additional information, such as the legal form, location, or the sector of a company.

Contributions and structure. We address the problem of automatically identifying the constituent parts of German company names by training a linear chain conditional random field classifier (CRF) [6], which uses features specifically designed for this task. In our evaluation, we analyze the impact of each feature on system performance. In particular, we make the following contributions:

- A tag-set used for marking the constituent parts of a company name
- A publicly available dataset of 1,500 segmented and labeled company names
- A system capable of identifying the constituent parts of company names with an accuracy of 84% and their colloquial names with an F1-measure of 88%

The following Section 2 discusses related work. In Section 3 we present our dataset and introduce the tag-set used for manually annotating the training and test data. We present the features and our selection strategy in Section 4. Finally, Section 5 presents our experimental results and Section 6 concludes the paper.

2 Related Work

Sequence labeling is a much addressed research topic [10] and is applied in a variety of contexts, such as part-of-speech tagging or the detection of gene substructures. A field in which sequence labeling is heavily employed is named entity recognition (NER) [5, 9]. Here, sequence labeling techniques are used to train classifiers to discover named entities, such as person or company names [7] in textual data.

A possible way to frame the problem of dissecting names into their constituent parts would be to formulate them as a nested named entity recognition (NNER) problem. In this setting, a company name represents a named entity whereas each constituent part of this entity is in turn a named entity on its own. A general approach for NNER was presented by Finkel and Manning [4]. To recognize nested named entities they employ a discriminative constituency parser that models sentences as trees. Byrne worked on historical archive texts and addressed the nested entity problem simply by generating concatenated tokens next to the original ones [1]. In contrast to those approaches, we focus on a specific solution for the domain of company names. We also need not deal with tree structures, because we assume that our input is always a single company name and not a sentence with arbitrarily deep structure. That is, we assume the outer loop of the nested NER is already performed.

Despite the good coverage of sequence labeling methods in research literature, very few publications focus on the issue of dissecting names into their constituent parts. A recent paper by Das et al. addresses this issue and focuses on the parsing of noisy person names obtained from LinkedIn, a website for professional networking [2]. Their approach must cope with the non-standardized order of constituent parts (first name, last name, . . .), as well as the occurrence of many optional ones (title, suffix, . . .).

As person names often occur as one of the many different constituent parts in a given company name, we regard the problem addressed by Das et al. as a sub-problem of decomposing company names. What makes the decomposition of company names particularly hard is that they display significant heterogeneity. Beside the different parts of person names, such as a title, company names contain additional components, such as locations, abbreviations, sector information, or other company details. This diversity results in the requirement of a larger tag set to distinguish the individual components of a given company name.

3 Labeling Name Components

In this section we introduce the dataset and tag-set used in our experiments, pointing out the specific difficulties encountered for each.

3.1 Dataset of company names

As our initial list of company names we extracted all company mentions from the Bundesanzeiger[3] website, an official gazette used by German federal agencies to publish company register changes. The obtained dataset comprises 796,389

[3] https://www.bundesanzeiger.de

company names of varying length and complexity. As such, the name spectrum ranges from very short names, such as “HV AG”, to very long ones with up to a hundred characters. We tokenize the company names by grouping character sequences into tokens and removing any whitespaces. Additionally, we merge identical tokens containing special characters and split compound words into their individual components by using the library JWordSplitter[4]. For example, this would turn the name “Autohaus Thomas Bloch e.K.” into the tokens [Auto][haus][Thomas][Bloch][e][.][K][.].

The average company name consists of 33 characters (median 30) and is 5 tokens long (median 4). To our surprise, we found that the dataset is remarkably clean, containing neither duplicates nor obviously false entries. The only problematic cases were names that the Bundesanzeiger website truncated at 100 characters. To avoid incomplete organization names, we irgnored all names with exactly 100 characters.

While the dataset contains mostly companies registered in Germany, it also contains organizations from other countries: Around 1.5% of the names belong to foreign companies, which we derived from their legal form, e.g., “HB” for Sweden or “ASA” for Norway. Interestingly, some names also include other company names, most often of their parent company, e.g., “Scheka Zweigniederlassung der Paul Sherwood Limited”.

We use 1,200 company names from the Bundesanzeiger as training data and 300 as test data. During the creation of the training set, we selected 155 foreign company names and filled the remaining 1,045 with a random sample to arrive at a properly diversified training set. We then tokenized and manually tagged all 1,500 company names.

3.2 Tagset

This section introduces the tag-set used for annotating each of the tokens of a company name. The proposed tagset was determined based on a qualitative analysis of 300 company instances. Each tag represents the semantic role within a given company name, for example the token [Berlin] would likely be assigned the tag Location. For a separate experiment we also tag each token that belongs to the colloquial name of a company.

Legal Form marks the legal form of a company. The most common legal form “GmbH” appears in almost 80% of company names. Most legal forms can be expressed in multiple ways. For example, “GmbH” is often written as “Gesellschaft mbH” or “Gesellschaft mit beschränkter Haftung”. Special characters, like “&” or periods, that represent a fixed part of a legal form were also tagged with this tag. Specializations that are not always part of the official legal form, like “haftungsbeschränkt” or “gemeinnützig” are also tagged as part of the legal form.

Person First Name & Person Last Name are used to mark the first/last name of a person. We annotate any middle names as first names. If a double

[4] https://github.com/danielnaber/jwordsplitter

name is connected via a hyphen, we tagged the hyphen as part of the name. Abbreviated names, such as the J in “J. Petschko Verwaltungs GmbH”, are tagged as first names only in clear cases, such as if it directly appears in front of a family name. Last names where treated accordingly.

Person Title tags any kind of title or degree a person can hold. Besides academic and nobility titles, such as “Doktor”, “Professor”, or “Prinz”, it is also used to tag religious titles, such as “St.” for saints.

Person Role handles all tokens concerning the position of a person within an organization. It is nearly always used to tag “Inhaber”/“Inhaberin” (owner) or “Apotheker” and is most often followed by a name or title.

Location is used for any kind of geo-information – most often cities or countries. Sometimes organization names also contain counties, regions, or street names and sometimes they particularize these information with cardinal directions. For instance, “Wohnungsunternehmen Nürnberg-Ost” specifies that the real estate company specializes in the eastern part of Nuremberg.

Proper Name. As the name suggests we use this tag to mark proper names, which are unique and synonyms of the company name. If in doubt during manual tagging, we checked the potential proper name by researching it on the web and only tagged the corresponding token if the results referred to the correct company. Examples include “Okabashi”, a manufacturer for sandals and flip flops, and “NIKE”, the sportswear manufacturer.

Punctuation is used to mark special characters that have no other role, including hyphens, quotation marks, periods, and exclamation marks.

Abbreviation marks tokens that are abbreviations for other parts of a name. In our context we regard an abbreviation only as such if the fully written version is also contained in the organization name. Considering the company name “AKoS - Akademie für Kompetenzentwicklung und Simulation GmbH”, the term “AKoS” would be labeled as an abbreviation.

Sector is used to annotate the industry sector of an organization. This also encompasses products or anything else that describes the activities of an organization. For example, the term “Wärmetechnik” would be included to describe that the company is concerned with heating systems.

Business Details. This tag is used to annotate everything that contains additional information about a company that is not already covered by one of the other tags. This encompasses common abbreviations like “i.L.”, which is attached to companies that are in liquidation, as well as information about how a company is structured, like “Partnerschaft” for partnerships or “Gebrüder” for companies that are lead by family members.

Other is the catch-all tag used for every token that could not be tagged otherwise.

Colloquial Name. In a separate experiment, we attempt to recognize a company’s colloquial name directly. In order to do this, we annotated not only the company components mentioned above, but also the terms which we thought of as a company's colloquial name. As a guideline we asked ourselves, under which name we would refer to the corresponding company in a newspaper article and annotated accordingly. For example consider the name "BVH Burkhard-vonHarder Pictures GmbH", in this case we marked "BHV Pictures" as the colloquial name.

3.3 Manual tagging for training

For the manual annotation process of the datasets we used the brat rapid annotation tool[5]. We pre-annotated unambiguous tokens using an ordered rule-based approach. The pre-annotation matches the token sequences of a rule with the tokenized organization names and, in case of a match, tags the matched tokens. This rule-based approach allowed us to pre-annotate about one third of the tokens, whereas we annotated the remaining tokens manually.

Table 1 shows how often which tag was used in the training and evaluation dataset. The occurrences and frequencies were counted once per token and once per name. The table shows that Legal Form and Sector are the most common tags. Person Title and Person Role occur fewer then fifty times in both datasets. This could pose a problem if the training dataset does not contain a good sample of the possible tokens that belong to those tags. It is interesting to note that nearly a third of all company names also contain a family name.

Tags	#tokens	%tokens	#names	%names
All	8,864	100.00%	1,500	100.00%
Legal Form	2,831	31.94%	1,484	98.93%
Business Details	91	1.03%	66	4.40%
Person First Name	181	2.04%	162	10.80%
Person Last Name	591	6.67%	488	32.53%
Person Title	24	0.27%	16	1.07%
Person Role	34	0.38%	25	1.67%
Location	344	3.88%	241	16.07%
Proper Name	229	2.58%	208	13.78%
Punctuation	998	11.26%	619	41.27%
Abbreviation	109	1.23%	86	5.73%
Sector	2,122	23.94%	1,116	74.40%
Colloquial Name	5,190	58.55%	1,500	100.00%

Table 1. Tag frequencies in the combined trainings and evaluation dataset.

4 Feature Generation & Selection

In this section we first focus on the engineering of our features and close by describing our feature selection process.

[5] http://brat.nlplab.org/

4.1 Feature set

Surface form features. One of the simplest and most effective token-based features is its surface form, which is the string of the token itself. To handle the many different word-forms of German words, we create an additional stemmed surface form feature that reduces the different word forms to their root. To this end, we used the Morphy German lemmatisation dictionary[6] and utilize a German snowball stemmer[7] for unmatched words. As our final surface feature, we use the surface form of the original compound word a token belongs to.

Prefix & suffix features. We use prefixes, suffixes, and their combination of different fixed lengths to create a number of features. Reducing a surface form to its suffix serves as a good indicator for the declination and conjugation without considering the word itself.

Positional features. Another useful feature is the positional index of a token within the entire company name: certain kinds of tags tend to appear at specific positions within a company name. For example, the Legal Form label often appears at the end of a name, whereas the Proper Name or Person First Name labels are more likely to be found at the beginning. As a consequence, we created three position features: The first represents the normalized position of the token. It ranges from 0 for the first token to 10 for the last. The other two features simply count the position of the token from the beginning and the end, respectively.

Shape features. Shape features try to capture the shape of a given word and place even unknown words into known, useful groups. One of the simplest shape features we used is the length of a surface, which serves as a good indicator for abbreviations and titles. A second feature is the "extended wordshape" feature, which replaces every single character of lower case letters with the single letter "x", every sequence of capital letters with a capital "X", numbers with a "#" and any other character with a "-". Therefore the extended shape of the word "AutoScout24" would be "XxxxXxxxx##". Another version of the word shape feature, called "condensed shape", replaces every sequence of lower case letters with the single letter of its shape representative. In this way, the word "AutoScout24" would be reduced to "XxXx#".

Context features. We designed context features that are capable of capturing relationships between a token and its context. Our first is a feature to identify abbreviations within company names. To determine whether a token could be an abbreviation, we split the token into sub-strings at each upper case letter. For instance, splitting the token [BeMiTec] in the name "BeMiTec Berlin Microwave Technologies Aktiengesellschaft" yields the strings "Be", "Mi" and "Tec". Next we check if these sub-strings match the first letters of other tokens in the name, which is indeed the case in the example.

[6] http://www.danielnaber.de/morphologie/

[7] http://snowball.tartarus.org/algorithms/german/stemmer.html

The second context feature is the “parenthesis feature”. It looks for a pair of opening/closing parentheses or quotation marks and assigns the opening character to every token between them.

The “named window shape feature” is inspired by the word shape feature but tries to capture the different token types within a window surrounding the current token. Within that window, each token containing letters is replaced by the letter “w” and each token containing digits by “#”. A token cannot contain both due to our tokenizer, which would produce two tokens in such a case. Subsequently, tokens that were originally separated by whitespace characters are separated by a single space whereas tokens which originally belonged to the same compound word are connected by an underscore. According to this approach, the window shape feature for the token [Carbon] in [3][C][Carbon][Group][GmbH] would yield “#_w w w w”.

The last context feature is the “word id” feature. We calculate it by dynamically creating a window around each token. The feature assigns the center token of the window the id 0. Each other token in the window is assigned its position relative to the center token. It counts the distance in words instead of tokens, which results in compound words receiving the same id. For example consider the window [Auto][haus][B][&][K]. Here the feature returns the labels -1 -1 0 1 2, because the tokens [Auto] and [haus] belonged to the same compound word.

Dictionary features. To improve the performance of specific labels we introduce a number of dictionary-based features. We use these dictionaries to find the longest possible token matches within a company name and mark each token, if it is part of a match.

The first dictionary is based on 350 regular expressions manually created from Wikipedia[8] to match a large number of international legal forms.

The next two dictionaries were created to help recognize first- and last-names. We created the first-name dictionary from DBpedia, which resulted in a dataset of 93,461 first names. The last-name dictionary was created from the German phone register dastelefonbuch.de[9] and contains 46,875 last-names.

We constructed a location dictionary by extracting the names of all OpenStreetMap objects of classes place or highway. The extracted places consist of cities and villages, but also countries, country regions and similar areas. Highways are all kinds of streets and paths. This dataset contains 157,972 unique names.

For capturing company sectors, we apply three different dictionary-based features: The first feature, “phone book sectors”, was created by extracting sector information from the Gelbe Seiten, a German business directory, resulting in a dictionary of 3,475 sector names. A second, improved version of the feature, called “sector SimString”, is based on sector names from Gelbe Seiten and DBpedia. The joint dictionary is comprised of 7,592 sector names. Instead of exact matching, this feature uses the SimString algorithm [11] to compute token matches.

[8] https://en.wikipedia.org/wiki/Types_of_business_entity

[9] http://www.dastelefonbuch.de/

The last two dictionary features use a different approach. Both try to build a dictionary of words which are commonly associated with sectors. The "sector keyword feature" does this by using a dictionary built from words commonly found in company sectors. To this end, we grouped all company names from Gelbe Seiten (Yellow Pages) and DBpedia by their industry sector. We tokenized the names and calculated the importance of each token using the double normalized tf-idf metric [12]. Every token with a tf-idf value over five, a length of at least two characters and at least one letter was used to create the dictionary. The sector description keyword classifier worked in a similar way, but used the DBpedia abstract of each sector as it's input.

4.2 Feature Selection

To measure the quality of our features, we used our training data to calculate their absolute and relative information gain. However, it turned out that the performance gained from using a feature often significantly diverges from what the calculated information gain suggests. Thus, selecting a good set of features for a classifier is of major importance. We publicly provide an overview over all features and their respective information gain together with our datasets[10].

In principle, any combination of features could be the best for a given classifier. Starting from our 41 features there are $2^{41}-1$ possible feature combinations, that need to be tested, so we apply a greedy approach: We start by using only the surface form feature as it is the most basic feature possible, and successively add those features that yield the maximal performance increase. We stop when no feature can be added that further increases the overall performance. While there is no guarantee that this greedy strategy results in an optimal solution, it significantly reduces the search space.

For our CRF classifier, the greedy approach yielded the feature combination ⟨surface form, absolute position, absolute position (rev.), long shape⟩, which we used during the training of the classifier.

5 Experiments

For the evaluation of our classifier we use the previously created test dataset consisting of 300 manually dissected and tagged names. During our experiments we use the Stanford implementation of a linear chain CRF classifier [3]. As is common practice, we withheld the test dataset while working on our features and during the training of the classifier. We trained the classifier using the feature set we described in Section 4.2 and the full training dataset of 1,200 company names. The detailed results of the evaluation are shown in Table 2, showing precision, recall, and F1-measure for each tag. To evaluate the overall classification quality, we also determined the accuracy by comparing the number of tags that were predicted correctly with the total number of tags.

[10] `https://hpi.de/naumann/projects/repeatability/datasets/company-name-dataset.html`

Tags	Precision	Recall	F1
Legal Form	98%	97%	98%
Business Details	75%	35%	48%
Person First Name	68%	56%	61%
Person Last Name	66%	69%	67%
Person Title	100%	60%	75%
Person Role	100%	55%	71%
Sector	86%	88%	87%
Location	62%	51%	56%
Proper Name	40%	40%	40%
Punctuation	91%	98%	94%
Abbreviation	78%	70%	74%
Overall accuracy	84%	–	–
Colloquial Name	88%	89%	88%

Table 2. Relative and absolute information gain per feature and the performance improvement of the CRF classifier

Although the classifier reaches an overall accuracy of 84% and very good results for several tags, especially for legal form and sector, there are clear differences in the classification performance when focusing on the individual tags, some of which we examine next.

In predicting the Legal Form we achieved the best overall F1-measure of 98%. The trainingset contains many examples of different legal forms, which enables the classifier to effectively train this class. Mistakes were made only in very untypical cases like the company name “ZPG mbH & Co. Z5 KG”, where the classifier labeled the period and the ampersand characters with the Punctuation tag.

The Business Details tag turned out to be one of the hardest to classify, because occurrences in our training set were quite rare and subject to much variation. Most errors occurred due to the confusion of business details and person names, such as the “Partner” token in the name “Xaver Scheingruber und Partner”. This decision is intuitively comprehensible, since in many cases the term “und” is followed by another persons name.

Misclassifications of the Person First Name and Person Last Name tags were mostly due to confusing them with proper names. Both are very similar regarding their structure. Often these terms are either unique or very infrequently used words that tend to start with a capital letter. It is difficult to distinguish whether “Xaven” or “Layden” are person or company names.

Person Title and Person Role tags were classified quite well. Both have a perfect precision score of 100%, the lower recall values of 60% and 55% can be explained with the high cardinality of both tags. For example, the title “Professor” occurred in the evaluation dataset but never in the trainingset and therefore was not recognized correctly. Such problems could be alleviated with larger training sets.

While the recall in recognizing the Sector tag was 88%, its precision was slightly lower and reached 85%. Here the “phone book sectors” feature was not as useful for identifying sector tokens as initially assumed, but it did help in the detection of colloquial names. The “sector SimString” feature, on the other hand, proved to be useful for determining the individual sectors.

The large variety of location names and their possible extensions make the Location tag a very hard problem for a classifier to solve. One idea would be to incorporate more sophisticated location services, such as ArcGIS[11] or Google Maps[12]. Usually, such services are able to distinguish the different components of a location, such as city and street name, which can then be used to further improve the results.

The classification of Proper Name tags resulted in our lowest performance. This is hardly surprising when regarding the enormous diversity of proper names. To improve the results, it could be promising to consult online services that could help to distinguish between individual classes. Another possibility would be to construct a significantly larger index than the one used in the text frequency feature.

For the classification of Abbreviation tags, misclassifications often occurred for single letter abbreviations, such as “G.A.S.”. A possible way to enhance the performance could be to extend the abbreviation feature by abbreviations that span multiple words and thereby trying to mitigate the issue.

For the evaluation of the Colloquial Name tag, we proceed in the same way as for the individual tags. With a precision of 88% and a recall of 89% the classification results are surprisingly good. In many cases, the errors are due to the fact that the algorithm tagged one token too much or too little. Occasionally, the classifier also selected a wrong name component for the colloquial name. For example in the case of the company “A.L.M. Europäische Freizeitsakademie GmbH” we marked “Europäische Freizeitsakademie” as the colloquial name, whereas the algorithm selected A.L.M.

We provide our labeled training data and evaluation results to the general public[13].

6 Conclusion and Future Work

We presented an algorithm capable of decomposing company names into their constituent parts, as well as identifying their colloquial forms. In addition, we defined a tagset that we used to manually annotate the constituent parts of a dataset consisting of 1,500 company names. Subsequently, we designed a large number of features that enabled us to successfully train a CRF-based classifier capable of identifying the constituent parts of a given company name. Using this set of features, we employed a greedy feature selection strategy to create a

[11] `http://www.esri.com/arcgis/about-arcgis`

[12] `https://maps.google.com`

[13] `https://hpi.de/naumann/projects/repeatability/datasets/company-name-dataset.html`

feature set for the training of our classifier. As a result, our classifier achieves an accuracy of 84% on classifying the constituent parts of a company name and 88% F1-measure in recognizing its colloquial name.

On the basis of our findings, some additional improvements could be explored. For instance, it is promising to include additional knowledge in form of external services, indexes, and dictionaries into the feature creation process. These knowledge sources could lead to a reduction of uncertainty during the classification process, which in turn would lead to a higher classification performance.

References

1. K. Byrne. Nested named entity recognition in historical archive text. In *IEEE International Conference on Semantic Computing (ICSC)*, 2007.
2. G. S. Das, X. Li, A. Sun, H. Kardes, and X. Wang. Person-name parsing for linking user web profiles. In *Proceedings of the ACM SIGMOD Workshop on the Web and Databases (WebDB)*, 2015.
3. J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*, 2005.
4. J. R. Finkel and C. D. Manning. Nested named entity recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2009.
5. R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 1996.
6. J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.
7. M. Loster, Z. Zuo, F. Naumann, O. Maspfuhl, and D. Thomas. Improving company recognition from unstructured text by using dictionaries. In *Proceedings of the International Conference on Extending Database Technology (EDBT)*, 2017.
8. L. C. Molina, L. Belanche, and À. Nebot. Feature selection algorithms: A survey and experimental evaluation. In *Proceedings of the International Conference on Data Mining (ICDM)*, 2002.
9. D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30, 2007.
10. N. Nguyen and Y. Guo. Comparisons of sequence labeling algorithms and extensions. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2007.
11. N. Okazaki and J. Tsujii. Simple and efficient algorithm for approximate dictionary matching. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, 2010.
12. G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24, 1988.
13. K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, 2003.