Labelling companies referred to in newspaper articles

Amine Nahid
Előd Egyed-Zsigmond
Sylvie Calabretto
amine.nahid@insa-lyon.fr
elod.egyed-zsigmond@insa-lyon.fr
sylvie.calabretto@insa-lyon.fr
Université de Lyon; LIRIS UMR 5205
Lyon, France

ABSTRACT

There are several domains where establishing links between newspaper articles and companies is useful. In this paper, we will present the first elements of our solution to predict links between a newspaper article written in French and a list of companies identified by their name and activity domain. We base our study on a semi-automatically annotated article corpus and the almost complete list of official French company names. We combine statistical linguistic methods with acronym generation and filtering techniques to propose a global score that predicts a distance between a text and a company. The main objective of the study presented in this paper is the creation of a usual name list for each company in order to improve the labelling of newspaper articles.

CCS CONCEPTS

• **Information systems** → **Document topic models**; *Relevance assessment*.

KEYWORDS

natural language processing, named entity recognition, information retrieval, text mining, text tagging

1 INTRODUCTION

Businesses have always had interest in assessing their performances, evaluating their financial and public relations situation. Hence, information contained in press articles, clients feed-backs, etc. might be of strategic importance.

Our main problem is to link articles with companies for a very large number of companies registered in France, and identified by their unique national identifier (SIREN code) and legal name. However, the companies are seldom referenced in the press using their legal names, that are often long. Our project is to design a solution to link economic press articles written in French with a set of companies. We have a semi-automatically annotated article ground truth corpus and the list of the official denominations of around 30,000 companies registered in France. Our main contribution in this paper is a protocol to construct the common names of companies given their legal name and the set of annotated articles.

We carry out our experiments and develop our tools on French language texts, but most of the methods used can be easily adapted to other languages.

This usual name list will then be combined to other methods to propose a global score that predicts a distance between a text and a company, in order to end up with a model that labels a newspaper article with its corresponding companies among our list.

2 STATE OF THE ART

Matching press articles with the companies they mention is part of the Named Entity Recognition (NER) domain. The term NER, appeared for the first time in the MUC-6 [5] conference. The task of recognising company mentions in texts is hence a sub-problem of NER, where we are interested only in entities representing companies. The issue can be addressed with different approaches. A baseline approach would be searching the official name of the company in the text. Nonetheless, searching the official name of a company within a newspaper article might reveal itself inefficient, given that most companies have usual or common names that slightly differ from their legal ones. Working on a German corpus, [3] proposed the use of dictionaries of colloquial names from various sources, as well as an alias generator that generates an alias out of an official denomination (it goes through some classic NLP data cleaning : removal of legal designations, special characters, geographic indications and token normalisation). There have been other works elaborating rule based systems, based on heuristics and/or hand crafted rules on a morphological level [4, 6, 7]. Unfortunately rule based methods are domain and language specific, and are not portable therefore. There are recently attempts to execute generic NER tasks, using deep learning [2], but they usually need much more training examples than we have, annotated more precisely. We are also experimenting with CRF (Conditional Random Fields) based techniques, with promising results. These experiments will be related in a future paper.

In the following section we propose a statistics based protocol to tackle the company recognition problem through common name dictionary generation.

3 PROPOSED APPROACH

In this section we present our company usual name creation method, first based on the official names and then on generated acronyms.

3.1 Hypothesis

Since companies are barely referred to by their legal names and are rather known by one or more common names, we need to provide an accurate automatic protocol to generate these common names.

Through the observation of the legal names of a set of French companies, we made the following hypotheses:

- The common name of a company might be only its legal name
- The common name of a company might be a contiguous sequence of terms that form the legal name (a sub word *ngram* of the legal name)
- The common name might be an acronym of the legal name or some part of it.

With these hypotheses, we aim to implement a common name generator that operates in two steps: as a first step it generates the sub-sequences and then the acronyms. The second step is the search of the best subset of *ngram*s and acronyms to compose the common name set.

3.2 Pre-processing

For our study, we have two data sets. The first one catalogues around 30k French companies identified by their SIREN codes (unique French identifier for businesses and not-for-profit organisations) and legal names in capital characters with no accents. The second data set contains around 120 thousand annotated French newspaper article URLs, manually labelled with the SIREN code of the companies they are talking about. Its elements are listed in accordance with the following scheme: id, SIREN code, legal name of the company, URL address of the article. We developed a scrapper that collected the title and body of the articles when available. We included finally only the articles for which we managed to scrap their content: title and text of the article. That gave us a dataset with around 58k articles.

We cleaned the official names from the first data set by removing the punctuation marks, especially the dots, commas and parentheses. However, we chose to keep the hyphens as their use in French is very common for compound names, considered as single terms in our model.
Examples:

- For companies without the special characters aforementioned, e.g. ELECTRICITE DE FRANCE, nothing is removed.
- For CA INDOSUEZ WEALTH (FRANCE), the parentheses are irrelevant and would be problematic for the *ngram* and *acronym* generation, the name has therefore got to be transformed to CA INDOSUEZ WEALTH FRANCE before any further process.
- There is a company registered in France with the official name: CASINO, GUICHARD-PERRACHON. The comma is to be removed (hence CASINO GUICHARD-PERRACHON) as it is useless for any future process. However, as written before, the hyphen is kept because GUICHARD-PERRACHON is actually one name and should not be considered as two separate terms.
- Also the dots are removed so as to normalise the acronyms in use within the legal names. e.g.: SARL and S.A.R.L.

For the second data set, we concatenate the titles and bodies under a unique attribute we called *corpus*. We also normalise the corpus by removing non-printable Unicode characters. The articles are then put into an *Elasticsearch* index.

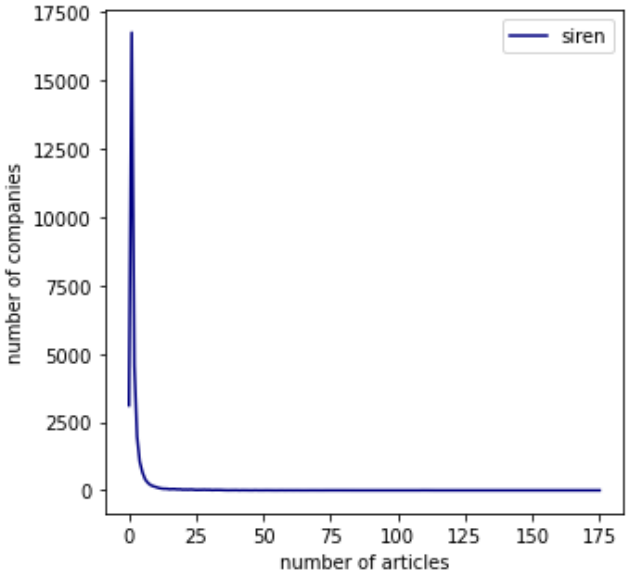

Figure 1: Number of companies per number of referencing articles

Since not all companies have the privilege of being talked about very often in the press, our ground truth shall be about the same. For our dataset, the graph (cf. Figure 1) shows the number of companies in function of the number of articles labeled as talking about them. 2375 French companies have more than 6 articles labeled as talking about them. We shall call these companies *well-documented companies* and focus our study on them. We consider that for the other $less - documented$ companies, it is difficult to generate usual names based on annotated articles.

3.3 *ngram* generator

We call *ngrams* all the contiguous sequences of terms contained in an expression. Our commitment at this is that for each company we generate all possible *ngrams* for its legal denomination.
For instance for the company *"COMPAGNIE DU RHONE"*, we should generate the following n-grams: *"COMPAGNIE", "DU", "RHONE", "COMPAGNIE DU", "DU RHONE", "COMPAGNIE DU RHONE".*

In order to filter potentially irrelevant *ngrams* we introduced 2 rules: filter one character long *ngrams*, filter *ngrams* based on their frequency in the official name list.

3.3.1 Occurrence frequency. For a given *ngram* we compute an inverse occurrence frequency score $of\_score(ngram)$ depending on the number of times it occurs in the company legal names set. The higher $of\_score(ngram)$ is, the more unique the n-gram is.

$$of\_score(ngram) = 1 - \frac{count(\{f \in C | contains(ln_f, ngram\}))}{count(C)} \tag{1}$$

Where:

- *ngram* is a subsequence of the legal name of the company c containing n words $0 < n \leq word\_count(ln_c)$

- ln_c, ln_f are the legal names of the companies c, f
- C is the set of all the companies we have
- $count(\{f \in C | contains(ln_f, ngram_c\}))$ is the size of the subset of companies from C containing $ngram_c$ in their official name

3.3.2 Threshold. Once we have the of_score for an *ngram*, we should define a threshold value $threshold_{ngram}$ that determines the *ngrams* to keep and those to discard according to their of_score. This is our second *ngram* filter.

$$If\ of\_score(ngram) \geq threshold_{ngram}\ then\ keep\ ngram \\ else\ discard\ it$$

After implementation and study we empirically set the $threshold_{ngram}$ value to 0.999.

At the end of this step, we end up with a key-value dictionary, called $dict_{ngram}$, where the keys are the SIREN codes of the companies and the values are lists of the potentially relevant *ngrams* for the given company.

3.4 Acronym generator

Many companies are referred to by an acronym in the press. We complete therefore our list of *ngrams* derived from the legal names with acronyms. For each legal name, we want to generate every potential acronym that might be in use. For instance:

- the SOCIETE FRANCAISE DU RADIOTELEPHONE is often referred to as ***SFR***
- SOCIETE DE DISTRIBUTION DE PAPIER would be more known as ***SODIPA***

The acronym generation is based on the first letter or the two first letters of each term of the whole legal name or a part of it.
Let us consider for example the company registered as TONNELLERIE FRANCOIS FRERES. Our acronym generator would provide the following possibilities:

- Based on the whole name: *TFF, TOFRFR, TOFRF, TOFFR, TFFR, TFRF, TFRFR, TOFF*
- Based on a part of the name: *TF, FF, FFR, FRF, FRF, TFR, TOF, TOFR*

Given our acronym generation protocol, for every company we should end up with a considerable amount of acronyms. More words a legal name has, the longer is the list of acronyms. We have therefore defined some rules to filter the potentially relevant ones.

- Any non-alphabetic symbol is removed, as part of data cleaning for this process.
- French stop words are removed from legal names before generating any acronym.
- No acronym generation for one-word long legal names.
- For each generated acronym, we verify in our ground truth of articles tagged with the concerned company whether there is at least one occurrence in the corpora. So, if an acronym occurs at least once then we keep it in our acronym dictionary, else we discard it.
- For performance reasons, no acronym is generated if the legal name is strictly more than 5 words long. 29 companies out of the list of 2357 $well-documented$ companies. For those few, we set their usual names manually.

Similarly to the previous step, we end up with a key-value dictionary, called $dict_{acr}$, where the keys are the SIREN codes of the companies and the values are lists of retained acronyms for the given companies.

3.5 F-measure

Having these two lists $dict_{ngram}$ and $dict_{acr}$ containing potential usual names, we have to create a method to keep those who are actually useful in order to link press articles to the companies. The first thing to do is to merge $dict_{ngram}$ and $dict_{acr}$ into a unique dictionary of "potential common names": $dict_{pcn}$. In this dictionary, every company, referred to by its SIREN code, has a list of unique *ngrams* and acronyms that have been retained after applying the filters aforementioned.
The following step is to generate for every "potential common name" list all the sub-sets. The aim is to select the sub-set that contains the most relevant common names for every company.
To do so, we compute the F-measure of every sub-list given our indexed ground truth. The latter contains articles from the second data set mentioned in the pre-processing part. For each article we know to which companies it refers to. We try to find one *ngram* set that we call $UsualNames_c$ for each company that maximises the F-measure when retrieving articles that contain at least one $ngram \in UsualNames_c$ when looking for a given company c.

The F-measure, introduced at the MUC-4 conference [1], is the harmonic mean of *precision* and *recall*.

We define as $relevant\_articles_c$ the articles that are annotated as talking about the company c in our dataset. We define as $retrieved\_articles_i$ the set of articles containing at least one of the common names in the i^{th} sub-set of the set of potential common names of a company c, as found in $dict_{pcn}(c)$ The *precision* is the ratio of $relevant\_articles_c$ amongst $retrieved\_articles_i$, whereas the *recall* is the ratio of *relevant articles* that were actually retrieved from $relevant\_articles_c$.
In fact, we query the press article Elasticsearch index, looking for articles that contain at least one of the *ngrams* contained the input *ngram* sub-set ($subset\_ngrams_c$). The documents that are returned upon the query are called $retrieved\_articles(subset\_ngrams_c)$. The ideal case would be that the query returns all articles about the company c the input sub-set belongs to; these are what we define as $relevant\_articles_c$.
e.g.: For BANQUE PALATINE let us suppose that we have, among others, the following potential common names sub-set: [PALATINE, BP], we would retrieve any articles containing PALATINE or BP at least once. Meanwhile the relevant articles are obtained with querying all the articles tagged with the SIREN code of BANQUE PALATINE in our ground truth dataset.

The formulae for the precision, the recall and the F-measure are:

$$F(subset_{c,i}) = 2.\frac{precision_{c,i} * recall_{c,i}}{precision_{c,i} + recall_{c,i}} \quad (2)$$

if $precision_{c,i} + recall_{c,i} \neq 0$ and $F(subset_{c,i}) = 0$ if $precision_{c,i} + recall_{c,i} = 0$

Where :

- $subset_{c,i}$ being the i^{th} sub-set of the set of potential common names of a company c, as found in $dict_{pcn}(c)$
- $0 < i \leq L_c$, L_c being the number of all the sub-sets of *ngrams* for the company c

Table 1: Mean values of the F-measure computation on the "well documented" companies using their common names (first row) and legal names only (second row)

	F-measure	Precision	Recall
best subset of common names	0.562	0.537	0.765
legal names only	0.446	0.476	0.560
difference	0.116	0.061	0.205

and

$$precision_{c,i} = \frac{|\{relevant\_articles_c\} \cap \{retrieved\_articles_i\}|}{|\{retrieved\_articles_i|\}} \quad (3)$$

$$recall_{c,i} = \frac{|\{relevant\_articles_c\} \cap \{retrieved\_articles_i\}|}{|\{relevant\_articles_c|\}} \quad (4)$$

Finally, once the computation is finished, we keep the subset that maximises the F-measure for every company as the definitive list of relevant common names of the concerned company.
The computation of the F-measures is important as the most relevant list of common names is not necessarily the largest one. In fact, adding an *ngram* or an *acronym* does not improve the F-measure in all cases. Sometimes, adding a new keyword (*ngram* or acronym) to a query might add noise to the results, and if the recall improves or stays constant, the precision might worsen and hence decrease the F-measure.

4 EXPERIMENTATION RESULTS

F-measures on $dict_{subpcn}$ and comparison with legal names

We ran the protocol described in the previous section on the 2357 $well - documented$ companies, i.e. those having at least 6 articles within the corpus labeled as talking about them.
We calculated $dict_{subpcn}$ as described in the previous section. We defined $dict_{first}$ which only keeps the subset that maximises the F-measure for each company, with its F-measure, precision and recall values.
We joined $dict_{first}$ with $dict_{flegal}$ to compare the F-measure of either the legal name or the best common name subset. Table 1 summarises the mean values of results obtained for this experiment.

- We obtained an average of 0.56 for the F-measure on the best common name subset for every company, whereas the average when tagging only through legal names for the same sample did not go beyond 0.44. Our protocol realised an average of 11.6% of F-measure improvement. It also improves the recall by 20.5%, which means that we retrieve more relevant articles using generated potential common names.
- The minimum difference of F-measure value is a negligible amount and can be considered as a nought value. We might affirm that our protocol can either improve things for article tagging or let them at a stable level.
- The precision improvement is quite low compared to the recall's improvement. This is mainly due to complex scrapping issues (such as URL redirections) that brought noise to the data, or omitted annotations on some articles. The latter issue is of interesting debate, should an article be labelled whenever it mentions a company, or only when the main topic is about it?

5 CONCLUSION AND PERSPECTIVES

In this study we proposed a method to build common, usual names for companies based on an official or legal name list including theirs and a large set of annotated news articles. Our experiments show that the method improves in a significant manner the F-score of the retrieval of relevant articles for a company when looking for our improved usual name set instead of only their official name. The filtering of the best common name sets through a greedy F-measure based approach is complete, but we can eventually try to give individual scores of each potential usual name and filter them according to that score, instead of studying all possible potential usual name set.

As our final goal is to label articles with companies they are talking about, we propose to work on other company-article distance measures. One is based on Part Of Speech tagging based NER (Named Entity Recognition) analysis combined with machine learning methods (for instance Conditional Random Fields) to predict words in a phrase that have a high probability of being company names. This can eliminate some false positives when company names are also common words like *Orange* or *But*. Another idea is to train a classifier to guess the activity sector an article is talking about and check whether it is close to the activity sector of the proposed companies. This method would help increasing the precision of our retrieval.

ACKNOWLEDGMENTS

To Infolégale, for providing the ground truth dataset.
To G. Benturquia, Y. Latreche, G. Meddour, T. E. Mekhalfa and A. E. Pereyra, for their most helpful work they have done during their fifth year research module.

REFERENCES

[1] Nancy Chinchor. 1992. MUC-4 evaluation metrics. In *Proceedings of the 4th conference on Message understanding (MUC4 '92)*. Association for Computational Linguistics, McLean, Virginia, 22–29. https://doi.org/10.3115/1072064.1072067
[2] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2018. A Survey on Deep Learning for Named Entity Recognition. *CoRR* abs/1812.09449 (2018). arXiv:1812.09449 http://arxiv.org/abs/1812.09449
[3] Michael Loster, Zhe Zuo, Felix Naumann, Oliver Maspfuhl, and Dirk Thomas. 2017. Improving Company Recognition from Unstructured Text by using Dictionaries. (2017), 10.
[4] Andrei Mikheev, Claire Grover, and Marc Moens. 1998. Description of the LTG System Used for MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*. Fairfax, Virginia. https://www.aclweb.org/anthology/M98-1021
[5] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticæ Investigationes* 30, 1 (Jan. 2007), 3–26. https://doi.org/10.1075/li.30.1.03nad
[6] L. F. Rau. 1991. Extracting company names from text. In *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application* (Miami Beach, FL, USA, 1991-02), Vol. i. IEEE, 29–32. https://doi.org/10.1109/CAIA.1991.120841
[7] GuoDong Zhou and Jian Su. 2002. Named Entity Recognition using an HMM-based Chunk Tagger. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 473–480. https://doi.org/10.3115/1073083.1073163