CEUR-WS.org/Vol-2847/paper-10.pdf

“This research is funded by...”: Named Entity Recognition of financial information in research papers

Daria Alexander[a], Arjen P. de Vries[b]

[a] *Spinque, Hooghiemstraplein 126 3514 AZ Utrecht, The Netherlands*

[b] *Radboud University, Faculty of Science ICIS office M1.00.05, Mercator 1 Toernooiveld 212 NL-6525 EC Nijmegen, The Netherlands*

Abstract

Customised Named Entity Recognition is an interesting, yet challenging task. The focus of our paper is the extraction of the named entities that provide financial information about research projects and programs. We introduce AckNER, a tool which extracts financial information from the “Acknowledgments” or “Funding” sections of research articles and dissertations. The results show that AckNER outperforms generic NLP libraries such as SpaCy, Stanza, FLAIR and DeepPavlov. The improvements found can be attributed to the ability to identify non-capitalised parts of the named entities in combination with the addition of patterns to extract information about contracts and grants.

Keywords

Named entity recognition, dependency parsing, regular expressions, named entity linking, knowledge graphs

1. Introduction

Named Entity Recognition is not a trivial task and is often domain-specific. One would not use the same algorithm for the recognition of the names of drugs and the names of politicians in old newspapers. That is why it is important to introduce algorithms adapted for a specific purpose.

In our study, we deal with information extraction from scientific articles and dissertations. The aim of our study is to understand which companies and organisations provide financial support for which research, what programs the research is part of and under which contract or grant this research is carried out.

Funding related information is important strategic management information in academia, for example to help understand which research output relates to the same research programs, and the subsidies that fund this type of research. Search tasks that involve this information are a specific case of enterprise search, common in large research organisations. Extracted named entities will be integrated into a Spinque knowledge graph (Spinque is a high-tech SME offering knowledge graph search technology), and linked to a number of external and internal databases of Delft University of Technology (TU Delft).

BIR 2021: 11th International Workshop on Bibliometric-enhanced Information Retrieval at ECIR 2021, April 1, 2021, online

daria@spinque.com (D. Alexander); arjen@cs.ru.nl (A. P. de Vries)

© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

For that purpose we designed AckNER, a tool that extracts the necessary financial information from the "Acknowledgements" and "Funding" sections of the article. The "Acknowledgements" section can include expected, if not imposed, acknowledgment of financial resources and research infrastructure, alongside very personal testimonies of gratitude [1]. More recent articles may contain a "Funding" section as well, in which case AckNER extracts information from it.

In this paper, we proceed as follows. First, we review research papers linked to our domains of interest. Then we present AckNER, the methodology and the evaluation procedure, followed by the results. Finally, we discuss the results and perspectives for future research.

2. Related work

Named Entity Recognition is the task of identifying named entities like person, location, organisation, drug, time, clinical procedure, biological protein in text, etc. [2]. Named Entity Recognition systems can be divided into 1) knowledge-based systems which are based on lexical resources [3], 2) bootstrapped systems, including orthography, context of the entity, words contained within named entities and also pattern extraction [4, 5], 3) feature-engineered supervised systems, that are using Hidden Markov Models, Support Vector Machines, decision trees and Conditional Random Fields [6, 7, 8] and 4) feature-inferring neural network systems, which are pre-trained on word and/or character embeddings [9, 10].

Different methods have been used for the extraction of named entities from a paper's "Acknowledgements" section. One way is to apply regular expressions [11]. Other methods include the usage of various pre-trained named entity recognisers such as `OpenCalais` and `AlchemyAPI` [12], `Stanford Core NLP` and `LingPipe` [13]. The recognisers are run in parallel and the overlapping named entities are eliminated. The extraction of named entities from the "Acknowledgements" shows that funding organisations receive more acknowledgement than any other category [11]. The information about funding organisations can be extracted using vocabularies, for example `CrossRef's Open Funder Registry` [13].

Named entities can also be extracted by using lexico-syntactic patterns [14, 15, 16]. Several studies have retrieved the named entities as tuples *(x,r,y)*, where x is the first entity, y is the second entity and r is the relation between them [17, 18]. POS-tagging can be the basis to identify patterns [17, 18]. However, using lexico-syntactic patterns is not always sufficient to identify the correct relationship between entities. Therefore, parsing trees that reflect the dependency relations between words are to be used [19]. Dependency parsing can create a link between two words in a tree even if they are far from each other in a sentence, due to the syntactic link between them and the head of the tree or sub-tree (usually a verb or a preposition) [20]. The named entity appearing first in the extracted dependency pattern is assumed to be the *effector* of the relation, usually a subject, while the second named entity is assumed to be the *effectee*, usually an object [21, 22].

Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities [23]. In a knowledge graph, the relationship between the entities is represented in the form of *(s,p,o)* triples - subject, predicate and object, - which is similar to the tuple issued from pattern extraction. Named entity linking is a tool to include named entities in the knowledge graph, where their meaning is disambiguated and unique identifiers can be assigned from an underlined knowledge repository [24]. Named entity recognition is performed before named entity linking, and errors in the first stage lead to poor performance during the second stage [25].

3. Methodology

AckNER is based on two supplementary methods: 1) dependency parse trees for extracting the names of the organisations, projects, programs and funds, and 2) regular expressions for retrieving information about contracts and grants. Contract and grant identifiers deserve special attention, as they provide essential information about funding.

Table 1
Dependency patterns

	Pattern	Example
Pattern 1	compound (one or more times) + object	**Danish/comp/National/comp/ Science/comp/Foundation/pobj/**
Pattern 2	modifier (one or more times) + coordinating conjunction(optional) + conjunction(zero or more times) + object	**863 /nummod/ High /nmod/ Technology/nmod/ and/cc/ Development/conj/ Project/pobj/**
Pattern 3	compound (one or more times) + object + of|for|on + compound(zero or more times) + object	**Netherlands /comp/ Institute /pobj/ for /prep/Space/comp/ Research/pobj/**

We have used the `spaCy` dependency parser to extract dependency patterns [26]. Table 1 shows the most common patterns extracted. We can see that all the patterns end with objects and can be preceded by compounds, modifiers, prepositions, conjunctions and a coordinating conjunction “`and`”. When the pattern contains a preposition “`of`”, “`for`” and “`on`” it can contain another object in the middle of the pattern.

These patterns were extracted from the sentences that contained various forms of the words “`fund`”, “`finance`” and “`support`” by filtering on their lemmas (base forms of the words). If we represent the sentences that contained the necessary named entities as tuples *(x,r,y)* we would notice that we need the third part of the tuple: y. The following examples represent the entities extracted with the patterns 1-3 in their context. X and r could also be used for the knowledge graph as a subject and predicate, but it is not the aim of our current task.

1. (This/det/ work/nsubjpass/, was/auxpass/ supported/root/ by/agent/, the/det/ *Danish/comp/ National/comp/ Science/comp/ Foundation/pobj/*)
2. (This /det/ work /nsubjpass/, was /auxpass/ supported /ROOT/ by /agent/, the/det/ *863 /nummod/ High /nmod/ Technology/nmod/ and/cc/ Development/conj/ Project/pobj/*)
3. (This/det/ project/nsubjpass/, is/auxpass/ funded/ROOT/ by/agent/, the/det/ *Netherlands /comp/ Institute /pobj/ for /prep/Space/comp/ Research/pobj/*)

To extract contract and grant identifiers, we use regular expressions to match patterns like *Contract No. DE-AC03-76-00098* and *grant No. BSIK03016.* Contract and grant information is also extracted from sentences that contain forms of the words “fund”, “finance” and “support”. For the extraction of contract numbers we use the following regular expression:

```
[Cc]ontract(No\.)?[A-Z0-9-]+
```

We extract the capitalised or non-capitalised word “contract” followed by the optional “No.” and by the combination of capitalised letters, numbers or dashes. The regular expression for the grant numbers is much alike:

```
[Gg]rant([Aa]greement)?(No\.)?[A-Z0-9-]
```

The word “agreement” may occur in the grant regular expression, otherwise the principle is the same. Overall, the dependency patterns and the regular expressions help to extract the named entities linked to financial information that are later used in the knowledge graph.

4. Evaluation

The evaluation data for testing AckNER consists of a random sample drawn from TU Delft’s institutional repository, containing 321 research articles and dissertations. These articles and dissertations were published between the 1980s and today. Out of 321 articles and dissertations, 102 contained an “Acknowledgments” or “Funding” sections. The articles and dissertations that did not contain “Acknowledgments” or “Funding” sections were mainly older scientific works.

To compare the results of our domain-specific approach to the results of generic NLP libraries, we have used four state-of-the-art libraries: SpaCy [26], Stanza (Stanford NLP Group) [27], FLAIR [28] and DeepPavlov [29]. Each of these NLP libraries uses pre-trained neural models. SpaCy uses a convolutional neural network (CNN) to train its models, FLAIR and Stanza pass their inputs to a bi-directional long-short term memory model (Bi-LSTM) and DeepPavlov uses a hybrid Bi-LSTM-CRF (conditional random field) model. For all the NLP libraries, we used packages that identify 18 entity types, such as location, organisation, money, *etc.* We decided to extract the named entities labelled as organisations (ORG). We also chose to extract the LAW entities from SpaCy, Stanza and FLAIR because some of the contract and grant numbers were labelled as LAW by those NLP libraries.

We have evaluated the results of extraction with AckNER and these four NLP libraries using a golden standard file that contained manually annotated named entities from the sample. We also took into account the difference between the annotation of AckNER and FLAIR and other NLP libraries. FLAIR and AckNER did not include the determiners “the” at the beginning of the words, although SpaCy, Stanza and DeepPavlov included them, so both variants were considered correct.

5. Results

Table 2 shows precision, recall and F1 measure for the selected NLP libraries and AckNER. We can see that AckNER significantly outperforms the generic NLP libraries, with a precision of

0.77 and a recall of 0.84. The high recall of AckNER is important to construct the knowledge graph used later, during search.

Table 2
Precision, Recall and F1-measure

NLP Libraries/algorithms	Precision	Recall	F1-measure
SpaCy	0.47	0.34	0.40
Stanza (Stanford NLP group)	0.51	0.39	0.43
FLAIR	0.47	0.41	0.44
DeepPavlov	0.63	0.46	0.53
AckNER	**0.77**	**0.84**	**0.80**

We can clearly see that the NLP libraries perform poorly on this data, especially in terms of recall. In our application, this low level of recall results in large numbers of named entities that would be missing from the knowledge graph. Among the NLP libraries, DeepPavlov performed the best, although it performs almost twice as poorly in terms of recall than AckNER. In the next section, we see what caused such results.

6. Discussion

The empirical results show that AckNER performs better than the selected NLP libraries. As for dependency patterns, one of the reasons for AckNER's higher performance is that it extracts the entities where some of the words are not capitalised; examples include *Smartmix funding program*, *ToKeN VindIT project*, *Dutch organization for Fundamental Research on Matter* or *BSIK / BRICKS project.* The NLP libraries only extract the entities or elements of entities that are capitalised (except prepositions and coordinating conjunctions); e.g., DeepPavlov and FLAIR recognise the entity *BSIK / BRICKS project* as *BSIK / BRICKS*, and SpaCy and Stanza miss it altogether.

Entities not extracted correctly by AckNER were complicated names of projects, e.g., *program "Smart systems based on integrated Piezo".* AckNER only retrieves *integrated Piezo* because, as the noun *program* is an object in the sentence, it terminates the previous pattern and starts looking for a new one. The same thing occurs to *project "Development of an Immersed Boundary Method, Implemented on Cluster and Grid Computers"*, from which only *Immersed Boundary Method* and *Cluster and Grid Computers* are extracted.

Some entities which contain the coordinating conjunction "and" are extracted properly; however, some others are not. For example, in the sentence *Peter Vajda was supported by the Slovak Research and Development Agency* the entity is *Slovak /nmod/ Research/nmod/ and/cc/ Development/conj/ Agency/pobj/*, which corresponds to our pattern. However, in the sentence *It was supported by Dutch Royal Academy of Arts and Sciences*, it is not extracted correctly. As the pattern always ends with an object, AckNER stops at *Arts* and does not extract *Sciences*, which is marked as a conjunction.

Another problematic extraction is linked to the fact that we retrieve all the sentences which contain various forms of the words "fund", "finance" and "support". AckNER extracts sentences like *We would like to thank G. Bihlmayer for technical support* and then retrieves all the entities that contain objects at the end, despite the fact that these sentences are not relevant.

As for contract and grant numbers, AckNER extracts 13 out of 14 named entities, one case of non-extraction being an error in the text pre-processing. FLAIR extracts three of these (two contract numbers and one grant number), Stanza retrieves two contract numbers, and SpaCy and DeepPavlov extract no relevant named entities. All the extractions of contract and grant numbers from Stanza and FLAIR are labelled as LAW. Although DeepPavlov does not have a LAW label or any other relevant label for extracting contract and grant numbers, it is perplexing that SpaCy, which has a LAW label, does not retrieve any relevant items. We can conclude that even when the NLP libraries have the label that is associated with the necessary information, this does not mean that they will retrieve all or even most of the expected named entities.

7. Conclusion

In this paper, we have explored the usage of the combination of dependency parsing patterns and regular expressions for the recognition of named entities in the "Acknowledgments"/"Funding" sections of research papers. We have focused on the entities that provide funding or financial information. We found that AckNER outperforms SpaCy, Stanza, FLAIR and DeepPavlov for two main reasons: 1) it extracts the entities that contained lowercase words and 2) it extracts almost all contract and grant numbers.

The main limitation of our study is that the sample size is small (102 articles that contain an "Acknowledgments"/"Funding" section, out of 321). However, as the NLP libraries did not recognise non-capitalised parts of named entities and extracted only a few contract and grant numbers, we do not expect the selected NLP libraries to outperform AckNER on a larger sample, and conclude that a domain-specific method like AckNER is better suited to our purpose.

The next steps of our research are to 1) run AckNER on a larger collection and 2) add extracted entities to a knowledge graph and link them to external and internal databases. Avenues for future work include using the small sample extracted for this paper as a starting point for semi-supervised learning, which will help us to add more and more relevant data to the sample and provide us with a large sample that will be added to the knowledge graph.

Acknowledgments

This work has been supported by the EU Horizon 2020 ITN/ETN on Domain Specific Sytems for Information Extraction and Retrieval (H2020-EU.1.3.1., ID: 860721).

8. Resources

Code and results are provided in GitHub repository https://github.com/informagi/AckNER.

References

[1] A. Paul-Hus, N. Descrochers, Acknowledgements are not just thank you notes: A qualitative analysis of acknowledgements content in scientific articles and reviews published in 2015, PLoS ONE 14 (2019). doi:https://doi.org/10.1371/journal.pone.0226727.

[2] V. Yadav, S. Bethard, A survey on recent advances in named entity recognition from deep learning models, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, p. 2145–2158.

[3] J. Callan, T. Mitamura, Knowledge-based extraction of named entities, in: CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management, 2002, pp. 532–537. doi:https://doi.org/10.1145/584792.584880.

[4] M. Collins, Y. Singer, Unsupervised models for named entity classification, in: P. Fung, J. Zhou (Eds.), Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP 1999, College Park, MD, USA, June 21-22, 1999, Association for Computational Linguistics, 1999, pp. 100–110. URL: https://www.aclweb.org/anthology/W99-0613/.

[5] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. Weld, A. Yates, Unsupervised named-entity extraction from the web: An experimental study, Artificial Intelligence 165 (2005) 91–134. doi:10.1016/j.artint.2005.03.001.

[6] R. Malouf, Markov models for language-independent named entity recognition, in: COLING-02: proceedings of the 6th conference on Natural language learning, volume 20, 2002, pp. 1–4. doi:https://doi.org/10.3115/1118853.1118872.

[7] X. Carreras, L. Màrquez, L. Padró, Named entity extraction using adaboost, in: COLING-02: proceedings of the 6th conference on Natural language learning, volume 20, 2002, pp. 1–4. doi:https://doi.org/10.3115/1118853.1118857.

[8] Y. Li, K. Bontcheva, H. Cunningham, Using uneven margins svm and perceptron for information extraction, in: CONLL '05: Proceedings of the Ninth Conference on Computational Natural Language Learning, 2005, pp. 72–79.

[9] R. Collobert, J. Weston, A unified architecture for natural language processing: deep neural networks with multitask learning, in: ICML '08: Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167. doi:https://doi.org/10.1145/1390156.1390177.

[10] Y. Kim, Y. Jernite, D. Sontag, A. M. Rush, Character-aware neural language models, AAAI (2016).

[11] C. Giles, I. Councill, Who gets acknowledged: Measuring scientific contributions through automatic acknowledgment indexing, Proceedings of the National Academy of Sciences of the United States of America 101 (2005) 17599–604. doi:10.1073/pnas.0407743101.

[12] M. Khabsa, P. Treeratpituk, C. Giles, Ackseer: A repository and search engine for automatically extracted acknowledgments from digital libraries, Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (2012). doi:10.1145/2232817.2232852.

[13] S. Kayal, Z. Afzal, G. Tsatsaronis, M. Doornenbal, S. Katrenko, M. Gregory, A Framework to Automatically Extract Funding Information from Text: 4th International Conference, LOD 2018, Volterra, Italy, September 13-16, 2018, Revised Selected Papers, 2019, pp. 317–328.

doi:10.1007/978-3-030-13709-0_27.

[14] M. A. Hearst, Automatic acquisition of hyponyms on large text corpora, in: COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics, 1992, pp. 539–545.

[15] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, A. Yates, Web-scale information extraction in knowitall: (preliminary results), in: WWW '04: Proceedings of the 13th international conference on World Wide Web, 2004, pp. 100–110. doi:https://doi.org/10.1145/988672.988687.

[16] C. Orna-Montesinos, Words and patterns: Lexico-grammatical patterns and semantic relations in domain-specific discourses, Revista Alicantina de Estudios Inglesese 24 (2011). doi:10.14198/raei.2011.24.09.

[17] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, S. Soderland, Textrunner: open information extraction on the web, in: Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Association for Computational Linguistics, Rochester, New York, USA, 2007, pp. 25–26.

[18] A. Fader, S. Soderland, O. Etzioni, Identifying relations for open information extraction, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK, 2011, p. 1535–1545.

[19] A. I. A. Aldine, M. Harzallah, B. Giuseppe, N. Béchet, A. Faour, Redefining hearst patterns by using dependency relations, in: Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, volume 2, 2018, pp. 148–155. doi:10.5220/0006962201480155.

[20] E. T. K. Sang, K. Hofmann, Lexical patterns or dependency patterns: Which is better for hypernym extraction?, in: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), Association for Computational Linguistics, Boulder, Colorado, 2009, p. 174–182.

[21] K. Fundel, R. Küffner, R. Zimmer, Relex—relation extraction using dependency parse trees, Bioinformatics 23 (2007) 365–371. doi:https://doi.org/10.1093/bioinformatics/btl616.

[22] H. Kilicoglu, S. Bergler, Syntactic dependency based heuristics for biological event extraction, in: Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, Association for Computational Linguistics, Boulder, Colorado, 2009, pp. 119–127.

[23] M. Krötzsch, G. Weikum, Journal of web semantics: Special issue on knowledge graphs, 2016.

[24] K. Balog, Entity-Oriented Search, volume 39 of *The Information Retrieval Series*, Springer, 2018.

[25] N. Botzer, Y. Ding, T. Weninger, Reddit entity linking dataset, 2021. Unpublished.

[26] Spacy, Spacy: Industrial-strength natural language processing in python, 2016. URL: https://spacy.io.

[27] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A python natural language processing toolkit for many human languages, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistic, 2020, pp. 101–108. doi:10.18653/v1/2020.acl-demos.14.

[28] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, R. Vollgraf, Flair: An easy-to-use framework for state-of-the-art nlp, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 54–59. doi:10.18653/v1/N19-4010.

[29] M. Burtsev, A. Seliverstov, R. Airapetyan, M. Arkhipov, D. Baymurzina, N. Bushkov, O. Gureenkova, T. Khakhulin, Y. Kuratov, D. Kuznetsov, A. Litinsky, V. Logacheva, A. Lymar, V. Malykh, M. Petrov, V. Polulyakh, L. Pugachev, A. Sorokin, M. Vikhreva, M. Zaynutdinov, Deeppavlov: Open-source library for dialogue systems, in: Proceedings of ACL 2018, System Demonstrations, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 122–127. doi:10.18653/v1/P18-4021.