Knowledge Extraction in Web Media: At The Frontier of NLP, Machine Learning and Semantics

Julien Plu
Supervised by Raphaël Troncy and Giuseppe Rizzo
Data Science Department, EURECOM, Sophia-Antipolis, France
julien.plu@eurecom.fr

ABSTRACT

We identify two main factors that can cause numerous difficulties when developing a generic entity linking system: i) the amount of data currently available on the Web that do not stop to increase and where a large part comes in the form of natural language texts; ii) the velocity at which data is published that may impose to process streams of text in near real-time. Social media platforms such as Twitter, Facebook or LinkedIn become a reliable source of news and play a key role for being aware of events around the world. Encyclopedia and newspaper articles contain general knowledge of our world and they can be used to explain concepts and known entities. Videos can be associated with subtitles and images may have captions. Depending on where a text comes from, it can have different properties such as a specific language, style of writing or topic. In this research, we present a preliminary framework based on a novel hybrid architecture for an entity linking system, that combines methods from the Natural Language Processing (NLP), information retrieval and semantic fields. In particular, we propose a modular approach in order to be as independent as possible of the text to be processed. Our evaluation suggests that this framework can outperform the state-of-the-art systems or show encouraging results on three datasets: OKE2015, #Micropost 2014 and #Micropost 2015. We identify the current limitations and we provide promising future research directions.

Keywords

Entity Extraction, Entity Linking, Entity Pruning, Adaptivity

1. PROBLEM

Entity recognition and entity linking in texts are two common tasks in the natural language understanding field. They are important for applications such as information extraction, content analysis, question answering or knowledge base population. The two main problems when working with natural language are ambiguity and synonymy, especially for entities. In this paper, we denote a *mention* as the textual surface form extracted from a text, an *entity* as a resource contained in a knowledge base, a *candidate entity* as a possible entity for a mention, and an *annotation* as a couple (mention, entity) defining a definitive link. We consider two different categories of texts: *i)* formal texts are well-written texts coming from trusted sources such a newspaper, magazine, or encyclopedia, *ii)* informal texts are texts coming from social media platforms or video subtitles. An entity may have more than one mention (synonymy) and a mention could denote more than one entity (ambiguity). For example, the mentions *HP* and *Hewlett-Packard* may refer to the same entity (synonymy), but the mention *Potter* can refer to many entities[1] (ambiguity). This problem can be extended to any language.

The task of entity linking is to annotate mentions extracted from a text to their corresponding entity contained in a knowledge base (KB). Each entry in the knowledge base represents uniquely a real world entity with a specific identifier. It is then useful to solve the problems of synonymy and ambiguity of natural language. An example of the entity linking task is represented in Figure 1.

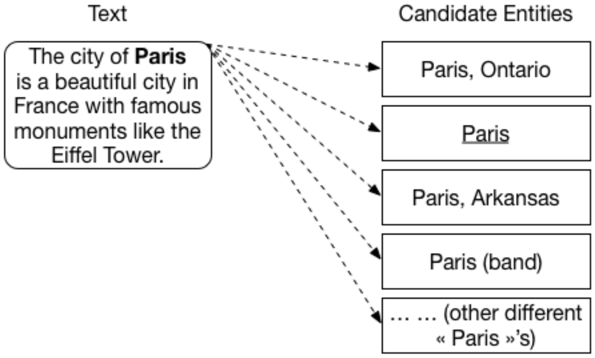
Text
Candidate Entities
The city of **Paris** is a beautiful city in France with famous monuments like the Eiffel Tower.
Paris, Ontario
Paris
Paris, Arkansas
Paris (band)
… … (other different « Paris »'s)

Figure 1: Figure representing an entity linking task. The mention extracted in the text is in bold and the correct entity is underlined among the candidate entities.

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
Include the http://dx.doi.org/10.1145/2872518.2888597

[1] https://en.wikipedia.org/wiki/Potter

Many knowledge bases can be used for doing entity linking: DBpedia[2], Freebase[3], Wikidata[4] to name a few. Those knowledge bases are known for being broad in terms of coverage, while vertical knowledge bases also exist in specific domains, e.g. Geonames[5], Musicbrainz[6] and LinkedMDB[7]. We can easily imagine that the results of an entity linking system highly depends of the knowledge base being used. For instance, if a text is about a movie and one only uses a geographical knowledge base (Geonames), the number of disambiguated entities is likely to be small in contrast to if a general purpose or cinema specific knowledge base is used. Emerging entities, i.e. entities that do not appear in the knowledge base being used, represent another problem. This phenomenon happens mainly in tweets where, for example, people that are just becoming popular and do not have yet an article in Wikipedia are mentioned.

The main research goal of this work is to define a solution to link entities depending on four parameters:

1. the nature of the text,
2. the language used,
3. the kind of entities to extract,
4. the knowledge bases to use

while processing either static documents or live streams and in a constrained time. Each of these parameters will be further explained in the Section 3. We identify two specific research questions:

RQ1 How to adapt an entity linking system depending on those parameters?

RQ2 How to design such a system in order to be able to process a large amount of data in near real-time?

In order to answer to these questions, we propose a hybrid system that aims to be as agnostic as possible of those parameters in a distributive and parallel manner. We have developed and analysed the performance of our system on three standard corpora. Results show that our approach can outperform existing systems.

2. STATE OF THE ART

An entity linking system can be divided in two parts: *Entity Extraction* and *Entity Linking*. We summarize the state-of-the art in the Tables 1 and 2, inspired by [1] and [15], where we show the similarities and differences of each system at extraction and linking level. In the last column of the Table 2, the abbreviation *LEE* stands for *Link Emerging Entities*.

For the extraction step, we observe that systems mainly use dictionaries based on a particular knowledge base (semantic-based approach). When POS tagging is being used, it is essentially a secondary feature which aims to enforce or to discard what has been extracted with the dictionary. Contrarily to the others, AIDA [6] uses a pure NLP approach based on Stanford NER [5]. TagME [4] claims to make an overlap resolution between the extracted mentions at the end of this process. Overlap resolution is the process of resolving at least two mentions that overlap in order to make just one mention using a defined heuristic. Further explanation about overlap resolution is provided in the next section. Our system tackles the problem using both a linguistic-based and a semantic-based approach of equal importance which demonstrates a higher performance at the extraction level.

For the linking step, we observe two approaches: graph-based (use the graph structure of the data) and arithmetic combination of functions (combining functions and then evaluate each function and then combine). Contrarily to the others, E2E [2] uses a pure machine learning approach using different features. At the end of this process, TagME [4] and WAT [12] do a pruning. None of these systems claim to be able to handle emerging entities, that is, disambiguate some entities to *NIL*. This is mainly due to their extraction approach. Our system tackles the problem using an arithmetic combination inspired from TagME [4] to detect and link emerging entities to *NIL* while using a pruning process at the end for removing the false positives.

3. PROPOSED APPROACH

Our goal is to link all the mentions occurring in a text to their entity counterparts in a knowledge base. Emerging entities will be linked to *NIL*. Although we are claiming that our approach is language and knowledge base independent, we present an evaluation based on English language texts where DBpedia 2014 is being used as knowledge base [14]. Formal texts are well formed and do not need to be normalized, unlike informal texts such as tweets that are notoriously problematic to process because of:

- hashtags (such as *#barackobama* referring to *Barack Obama*)
- user mentions (e.g. *@ryeong9* referring to *db:Ryeo_Wook*[11])
- acronyms (e.g. *Met* for *Metropolitan Police Service*)
- short length of only 140 characters
- syntax which is often grammar free with misspelled words

Our approach is divided in two main steps, entity extraction and entity linking, and two optional ones, text normalization and entity pruning. In the following, we briefly describe these different components.

Text Normalization. Informal texts are normalized by removing emoticons, extra white spaces and punctuation symbols belonging to two unicode categories[12]: other and symbol.

Entity Extraction. This step is about detecting mentions from text (formal and informal) that are likely to be selected as entities. It is composed of two different modules: *extractors* and *overlap resolution*. The objective of the extractors module is achieved by using the five following components: POS tagger for noun phrases, POS tagger for numbers, NER, dictionary, and time expressions spotter. For informal texts, we add a dereferencing social media account component to retrieve the corresponding user name. Each of these component can be activated or deactivated depending on the kind of entities to extract. We use the Stanford NLP POS Tagger. For informal texts, we use it with a model trained specifically for tagging tweets[13] in order to be case insensitive and to get independent tags for mentions and hashtags. For formal texts written in English, we use

[2] http://wiki.dbpedia.org

[3] https://www.freebase.com

[4] https://www.wikidata.org/

[5] http://www.geonames.org

[6] https://musicbrainz.org

[7] http://www.linkedmdb.org

[8] Dual Coordinate Descent-Structural Support Vector Machine

[9] Multiple Additive Regression Trees

[10] Term Frequency*Inverse Corpus Frequency

[11] db stands for http://dbpedia.org/resource/

[12] http://www.fileformat.info/info/unicode/category/index.htm

[13] https://gate.ac.uk/wiki/twitter-postagger.html

EE (Entity Extraction)				
System	External Tools	Main Features	Method	Knowledge Base
E2E [2]	-	N-Grams, stop words removal, punctuation as token	rule-based (candidate filter), dictionary	Wikipedia, Freebase
AIDA [6]	StanfordNER	-	-	NER (Named Entity Recognition) dictionary
TagME [4] and DataTXT [16]	-	N-Grams, overlap resolution, Wikipedia statistics	dictionary, link probability	Wikipedia
WAT [12]	OpenNLP	N-Grams, Wikipedia statistics	dictionary, SVM (Support Vector Machine)	Wikipedia, NER dictionary
Babelfy [11]	-	N-Grams, POS (Part-of-Speech), superstring matching	dictionary	Babelnet
Spotlight Lucene [9]	LingPipePOS	string matching, POS	Aho-Corasick, dictionary	DBpedia
Spotlight statistical [3]	OpenNLP	POS, capitalized words	finite-state automaton, Aho-Corasick	DBpedia, NER dictionary
VINCULUM [8]	StanfordNER	-	-	NER dictionary
FOX [17]	StanfordNER, OpenNLP, Illinois NE Tagger, Ottawa Baseline IE	-	ensemble learning using 15 different classifiers	NER dictionary

Table 1: Entity Extraction analysis

EL (Entity Linking)				
System	Main Features	Method	Knowledge Base	LEE
E2E [2]	N-Grams, lower case, entity graph features, popularity based on clicks and visiting information on the Web	DCD-SSVM[8] + MART[9] gradient boosting	Wikipedia, Freebase	No
AIDA [6]	popularity based on Wikipdia, similarity, coherence, densest subgraph	graph-based	YAGO2	No
TagME [4] and DataTXT [16]	mention-entity commonness, Wikipedia statistics	collective agreement, link probability, C4.5	Wikipedia	No
WAT [12]	string similarity, commonness, context similarity, PageRank, Personalized PageRank, HITS, SALSA	graph-based	Wikipedia	No
Babelfy [11]	densest subgraph	graph-based	Babelnet	No
Spotlight Lucene [9]	TF*ICF[10]	VSM (Vector Space Model), cosine similarity	DBpedia	No
Spotlight Statistical [3]	popularity based on Wikipedia, string similarity, context similarity	multiplication among the three features, best result is taken	DBpedia	No
VINCULUM [8]	types, co-reference, coherence	sum maximisation among the coherence scores	Wikipedia	No
FOX [17]	HITS	graph-based	DBpedia	No

Table 2: Entity Linking analysis

the *english-bidirectional-distsim* model that provides a better precision but for a higher computing time. We rely on Stanford NER that we properly re-trained with the training set of a given benchmark dataset. The dictionary reinforces this stage bringing a robust spotting for well-known proper nouns. Each of those components are launched in parallel. The type of an entity is given by Stanford NER. If a mention is not detected by Stanford NER and then linked to *NIL*, no type is assigned to it. Each of these components can extract mentions that have a partial or a full overlap with others. To solve this problem, we implement an *overlap resolution* module that takes the output of each component of the extractors module and gives one output without overlaps. The logic of this module if the following: given two overlapping mentions, e.g. *States of America* from Stanford NER and *United States* from Stanford POS tagger, we only take the union of the two phrases. We obtain the mention *United States of America* and the type provided by Stanford NER is selected.

Entity Linking. This step is composed of three subtasks:

1. entity generation, where we lookup up the entity in an index built on top of both DBpedia2014[14] and a dump of the Wikipedia articles[15] dated from October 2014 to get possible entity candidates;
2. entity candidates filtering based on direct inbound and outbound links in Wikipedia between the entity candidate of each mention;
3. entity ranking based on an in-house ranking function $r(l)$

$$r(l) = (a{\cdot}L(m, title) + b{\cdot}max(L(m, R)) + c{\cdot}max(L(m, D))) \cdot PR(l) \tag{1}$$

The function $r(l)$ is using the Levenshtein distance L between the mention m and the title, the max distance between the mention m and every element (title) in the set of Wikipedia redirect pages R and the max distance between the mention m and every element (title) in the set of Wikipedia disambiguation pages D, weighted by the PageRank PR, for every entity candidate l. The weights a, b and c are a convex combination that must satisfy: $a + b + c = 1$ and $a > b > c > 0$. We take the assumption that the string distance measure between a mention and a title is more important than the distance measure with a redirect page which is itself more important than the distance measure with a disambiguation page. If an entity does not have an entry in the knowledge base, we normally link it to *NIL*.

Entity Pruning. This step is used to detect and remove the false positive annotations in order to improve the precision of the system. We use a machine learning approach, with the algorithm k-NN. We have tried four different algorithms (Random Forest, Naive Bayes, SVM and k-NN), and have empirically assessed that k-NN generally provides the best results. To train this algorithm and to get a model, we use ten features: the mention, title, type, PageRank, HITS, inLinks, outLinks, length of the article, number of redirects, from the index, and $r(l)$ of the entity. The training method uses a four steps approach:

1. run our system on a training set;
2. classify annotations as true (should appear in the results) or false (should not appear in the results) according to the entities in the Gold Standard of the training set and the ones provided by the results of our system;
3. create a file with the features of each of these entities and their true / false classification;
4. train k-NN with this file that contains the features to get a model.

Once the model has been created, we let k-NN classify each annotation provided by our system to true or false.

What makes this approach adaptive is that each step corresponds to at least one parameter described in the Section 1. *Text Normalization* is agnostic to the language. *Entity Extraction* can also be agnostic to the language providing that the POS Tagger and the NER can use a model trained over a specific language together with a specific dictionary. It is also agnostic to the type of entities to extract as we can train the NER to recognize particular types. *Entity Linking* is agnostic (for now) to any DBpedia as it uses an index built with shared properties among each DBpedia version. Finally, the entire system is itself agnostic to the kind of text as it can be adapted according to the text we have to process.

4. METHODOLOGY

The methodology used in the development of this work comprises three tasks:

1. Systematic review of the state of the art: this includes the study of the literature about Named Entity Linking approaches published in the Natural Language Processing, Semantic Web and Information Retrieval fields.
2. Formalization of the problem to solve and formulation of research questions and hypothesis. We have then proposed a solution identifying some novel contributions. We have implemented the proposed framework that we aim to release as open source to the community.
3. Evaluation of the proposed framework to measure its performance with respect to the other state of the art approaches. The experiments will be conducted as follows:

   (a) definition of benchmarks on known datasets to measure the quality of our approach;

   (b) execution of experiments and statistical studies of the obtained results to deduce conclusions about the proposed solution. The measures we will use are: precision, recall and F-measure;

   (c) error analysis to understand why some specific errors occurred.

   This three step evaluation process will be used for every new implementation of the proposed solution.

5. RESULTS

Our hybrid approach has been tested against the test dataset of the #Micropost2014 [1] and #Micropost2015 [15] NEEL challenges and the OKE2015 challenge[16]. The breakdown results for each of these datasets are available[17]. Table 4 shows the performance of our approach in comparison with state of the art systems given the F1-measure at the final linking stage. The results of our approach are given without the pruning step as it is still experimental.

Results of challenges come from the official published results. We report on the best performing systems: E2E, DataTXT, AIDA, UTwente for #Micropost2014 and ousia, acubelab for #Micropost2015. The results of our approach for those two challenges have been computed with the official scorer used by the organizers. For the OKE2015 challenge, that we won [13], we have, afterwards, changed the test dataset in order to fix annotation issues and those changes have been approved by the organizers and committed to the official repository. We used the neleval[18] scorer instead of the official one used in the challenge. The results

[14] http://wiki.dbpedia.org/services-resources/datasets/datasets2014

[15] https://dumps.wikimedia.org/enwiki/

[16] https://github.com/anuzzolese/oke-challenge

[17] http://multimediasemantics.github.io/adel/

[18] https://github.com/wikilinks/neleval

Datasets	co-references	typing	emerging entities	dates	numbers
OKE2015	✓	✓	✓	✗	✗
#Microposts2014	✗	✗	✗	✓	✓
#Microposts2015	✗	✓	✓	✗	✗

Table 3: Features for each datasets

	Our Approach	FOX	DataTXT	FRED	AIDA	E2E	UTwente	ousia	acubelab
#Microposts2014	46.29	*N/A*	49.9	*N/A*	45.37	**70.06**	54.93	*N/A*	*N/A*
#Microposts2015	47.95	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	**80.67**	47.57
OKE2015	**60.75**	49.88	*N/A*	34.73	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*

Table 4: F1-measure results at the linking stage on the #Microposts2014 and 2015 NEEL challenge, and OKE2015 challenge test datasets

of this new evaluation is also available[19]. These differences explain why the results changed compared to the ones reported by the organizers of the challenge. Those datasets have differences listed in Table 3. We have chosen those different datasets to show the full potential of our approach which can be adapted depending on the dataset to be processed in terms of features and kind of text.

For formal text (OKE benchmark), our system outperforms the other approaches being tested. For informal text (NEEL corpora), the approach shows the robustness in extracting and typing entities because we jointly use linguistic and semantic methods. The results show a big drop at linking level, mainly due to a unsupervised method for entity linking.

6. CONCLUSIONS AND FUTURE WORK

The results are encouraging since we demonstrate that this approach, which exploits both linguistic and semantic features, enables to achieve:

- language adaptivity: another language than English can be used by changing the language of DBpedia, the models used by Stanford Core NLP and the surface forms that the dictionary may contain;
- text adaptivity: different kind of text (newswire, tweets, blog posts, etc.) can be processed.

Furthermore, our approach enables partially to achieve:

- knowledge base adaptivity: different DBpedia knowledge bases (in terms of language and version) can be indexed;
- entity adaptivity: although focusing on common types (PERSON, LOCATION and ORGANIZATION), dates and numbers can also be independently extracted and linked.

Our approach is not yet capable of handling live streams of text due to its excessive processing time. As future work, we plan to improve the linking, the pruning and the overall architecture to scale up.

Linking. The linking step is currently the main bottleneck in our approach. The performance drops significantly at this stage mainly due to a fully unsupervised method. Two new methods can be investigated in order to improve this step. The first one consists in replacing the Levenshtein distance by the word2vec similarity in the formula used by our approach. The word2vec [10] algorithm will be trained with Wikipedia anchors and titles. The second method is to use the Deep Semantic Relatedness Model [7] as a relatedness score between each candidate to build a graph composed of these candidates where each edge is weighted by this score. The path that has the highest score is chosen as the good one to disambiguate each extracted entity. Other general knowledge bases such as Freebase and Wikidata will be tested, but also specific ones like Geonames and LinkedMDB for different kind of text in order to broaden the evaluation domain of our approach. Finally, to better handle emerging entities, a *NIL* clustering module will be developed in order to group together the mentions that may represent the same unknown entity and to assign a type accordingly to a contextual meaning in the text.

Pruning. The pruning step shows encouraging results but is still far from reaching its full potential. In order to improve this stage, three methods can be investigated. The first method is to select better features with a stronger classifier algorithm like SVM. In order to get better features, we can either select them manually or using a feature learning algorithms such as RBM[20]. The second method is to use an ensemble learning by using multiple classifier algorithm (k-NN, SVM, Naive Bayes, C4.5...) and combine them with an algorithm that would select the best result among those multiple classifiers. The third method is to use deep learning algorithms such as CNN[21] or RNN[22]. While RNN fits better NLP tasks, CNN can sometimes provide good performance for those tasks and has to be investigated as well.

Architecture. The current architecture is insufficiently extendable. Adding external tools such as OpenNLP is not trivial and will not scale to provide results on live streams of text (initial test shows a response time that varies between 2 and 6 seconds per document). The current system is not designed to be distributed and parallelization could be worked further. The approach has three main bottlenecks: i) the index creation that takes around 2 days for a large knowledge base, ii) the index lookup that could be optimized and iii) the repetitive usage of Stanford NLP that constantly loads the same models. We aim to rely on CouchBase[23] and Elasticsearch[24] to tackle the first two bottlenecks, with the goal of getting a distributed system that can run on a cloud infrastructure. We plan to develop a generic API to not only rely on Stanford Core NLP but any other NLP toolkits. Next, in order to handle live streams, a solution such as Apache Spark[25], plugged on top of the system will be investigated.

[19] http://multimediasemantics.github.io/adel/

[20] Restricted Boltzmann Machines

[21] Convolutional Neural Network

[22] Recurrent Neural Network

[23] http://www.couchbase.com

[24] https://www.elastic.co/products/elasticsearch

[25] http://spark.apache.org/

Acknowledgments

This work was partially supported by the innovation activity 3cixty (14523) of EIT Digital.

7. REFERENCES

[1] A. E. C. Basave, G. Rizzo, A. Varga, M. Rowe, M. Stankovic, and A. Dadzie. Making sense of microposts (#microposts2014) named entity extraction & linking challenge. In *4th Workshop on Making Sense of Microposts*, Seoul, Korea, 2014.

[2] M. Chang, B. P. Hsu, H. Ma, R. Loynd, and K. Wang. E2E: an end-to-end entity linking system for short and noisy text. In *4th Workshop on Making Sense of Microposts*, Seoul, Korea, 2014.

[3] J. Daiber, M. Jakob, C. Hokamp, and P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *9th International Conference on Semantic Systems, (I-SEMANTICS)*, Graz, Austria, 2013.

[4] P. Ferragina and U. Scaiella. TAGME: on-the-fly annotation of short text fragments (by wikipedia entities). In *19th ACM Conference on Information and Knowledge Management, (CIKM)*, Toronto, Ontario, Canada, 2010.

[5] J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *43rd Annual Meeting of the Association for Computational Linguistics*, 2005.

[6] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. Robust disambiguation of named entities in text. In *Conference on Empirical Methods in Natural Language Processing, (EMNLP)*, Edinburgh, UK, 2011.

[7] H. Huang, L. Heck, and H. Ji. Leveraging deep neural networks and knowledge graphs for entity disambiguation. *CoRR*, 2015.

[8] X. Ling, S. Singh, and D. S. Weld. Design challenges for entity linking. *TACL*, 2015.

[9] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *7th International Conference on Semantic Systems,(I-SEMANTICS)*, Graz, Austria, 2011.

[10] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, 2013.

[11] A. Moro, A. Raganato, and R. Navigli. Entity linking meets word sense disambiguation: a unified approach. *TACL*, 2014.

[12] F. Piccinno and P. Ferragina. From tagme to WAT: a new entity annotator. In *First ACM International Workshop on Entity Recognition & Disambiguation, (ERD)*, Gold Coast, Queensland, Australia, 2014.

[13] J. Plu, G. Rizzo, and R. Troncy. A Hybrid Approach for Entity Recognition and Linking. In *12^{th} European Semantic Web Conference, Open Knowledge Extraction Challenge*, 2015.

[14] J. Plu, G. Rizzo, and R. Troncy. Revealing entities from textual documents using a hybrid approach. In *3rd International Workshop on NLP & DBpedia*, Bethlehem, Pennsylvania ,USA, 2015.

[15] G. Rizzo, A. E. C. Basave, B. Pereira, and A. Varga. Making sense of microposts (#microposts2015) named entity recognition and linking (NEEL) challenge. In *5th Workshop on Making Sense of Microposts*, Florence, Italy, 2015.

[16] U. Scaiella, M. Barbera, S. Parmesan, G. Prestia, E. D. Tessandoro, and M. Verí. Datatxt at #microposts2014 challenge. In *4th Workshop on Making Sense of Microposts*, Seoul, Korea, 2014.

[17] R. Speck and A. N. Ngomo. Ensemble learning for named entity recognition. In *13th International Semantic Web Conference, (ISWC)*, Riva del Garda, Italy, 2014.