Using a Hybrid Approach for Entity Recognition in the Biomedical Domain

Basaldella, Marco ; Furrer, Lenz ; Colic, Nicola ; Ellendorff, Tilia Renate ; Tasso, Carlo ; Rinaldi, Fabio

Abstract: This paper presents an approach towards high performance extraction of biomedical entities from the literature, which is built by combining a high recall dictionary-based technique with a high-precision machine learning filtering step. The technique is then evaluated on the CRAFT corpus. We present the performance we obtained, analyze the errors and propose a possible follow-up of this work.

Using a Hybrid Approach for Entity Recognition in the Biomedical Domain

Marco Basaldella
Università degli Studi di Udine
Via delle Scienze 208, Udine
basaldella.marco.1@spes.uniud.it

Lenz Furrer
Institute of Computational Linguistics
University of Zurich
Andreasstrasse 15, CH-8050 Zürich
lenz.furrer@uzh.ch

Nico Colic
Institute of Computational Linguistics
University of Zurich
ncolic@gmail.com

Tilia R. Ellendorff
Institute of Computational Linguistics
University of Zurich
ellendorff@cl.uzh.ch

Carlo Tasso
Università degli Studi di Udine
carlo.tasso@uniud.it

Fabio Rinaldi
Institute of Computational Linguistics
University of Zurich
fabio.rinaldi@uzh.ch

Abstract

This paper presents an approach towards high performance extraction of biomedical entities from the literature, which is built by combining a high recall dictionary-based technique with a high-precision machine learning filtering step. The technique is then evaluated on the CRAFT corpus. We present the performance we obtained, analyze the errors and propose a possible follow-up of this work.

1 Introduction

The problem of technical term extraction (herein TTE) is the problem of extracting relevant technical terms from a scientific paper. It can be seen as related to Named Entity Recognition (NER), where the entities one wants to extract are technical terms belonging to a given field. For example, while in traditional NER the entities that one is looking for are of the types "Person", "Date", "Location", etc., in TTE we look for terms belonging to a particular domain, e.g. "Gene", "Protein", "Disease", and so on (Nadeau and Sekine, 2007). A further evolution is the task of Concept Recognition (CR), where the entity is also matched to a concept in an ontology.

NER (and then TTE) can be solved using very different techniques:

- Rule-based approach: a group of manually written rules is used to identify entities. This technique may require deep domain and linguistic knowledge. A simple example may be the task of recognizing US phone numbers, which can be solved by a simple regular expression.
- Machine learning-based approach: a statistical classifier is used to recognize an entity, such as Naive Bayes, Conditional Random Fields, and so on. Several different types of features can be used by such systems, for example prefixes and suffixes of the entity candidates, the number of capital letters, etc. A major drawback of this approach is that it typically requires a large, manually annotated corpus for algorithm training and testing.
- Dictionary-based approach: candidate entities are matched against a dictionary of known entities. The obvious drawback of this approach is that it is not able to recognize new entities, making this technique ineffective e.g. in documents which present new discoveries.
- Hybrid approaches: two or more of the previous techniques are used together. For example, Sasaki et al. (2008) as well as Akhondi et al. (2016) combine the dictionary and ML-based approaches to combine the strengths of both.

The aim of this work is to propose a hybrid approach based on two stages. First, we have a dic-

tionary phase, where a list of all the possible terms is generated by looking for matches in a database. This aims to build a low precision, high recall set with all the candidate TTs. Then, this set is filtered using a machine learning algorithm that ideally is able to discriminate between "good" and "bad" terms selected in the dictionary matching phase to augment the precision.

This approach is realized by using two software modules. The first phase is performed by the OntoGene pipeline (Rinaldi et al., 2012b; Rinaldi, 2012), which performs TTE from documents in the biomedical field, using a dictionary approach. Then, OntoGene's results are handed to Distiller, a framework for information extraction introduced in Basaldella et al. (2015), which performs the machine learning filtering phase.

2 Related Work

The field of technical term extraction has about 20 years of history, with early works focusing on extracting a single category of terms, such as protein names, from scientific papers (Fukuda et al., 1998). Later on, "term extraction" became the common definition for this task and some scholars started to introduce the use of terminological resources as a starting point for solving this problem (Aubin and Hamon, 2006).

While the most recent state-of-the-art performance is obtained by using machine learning based systems (Leaman et al., 2015), there is growing interest in hybrid machine learning and dictionary systems such as the one described by Akhondi et al. (2016), which obtains interesting performance on chemical entity recognition in patent texts. In the field of concept recognition, there are different strategies for improving the coverage of the recognized entities. For example, known orthologous relations between proteins of different species can be exploited for the detection of protein interactions in full text (Szklarczyk et al., 2015). Groza and Verspoor (2015) explore the impact of case sensitivity and the information gain of individual tokens in multi-word terms on the performance of a concept recognition system.

The CRAFT Corpus (Bada et al., 2012) has been built specifically for evaluating this kind of systems, and is described in detail in Section 3.1. Funk et al. (2014) used the corpus to evaluate several CR tools, showing how they perform on the single ontologies in the corpus. Later, Tseytlin et al. (2016) compared their own NOBLE coder software against other CR algorithms, showing a best F1-score of 0.44. Another system that makes use of CRAFT for evaluation purposes is described in Campos et al. (2013).

3 System Design

3.1 CRAFT Corpus

The CRAFT corpus is a set of 67[1] manually annotated journal articles from the biomedical field. These articles are taken from the PubMed Central Open Access Subset,[2] a part of the PubMed Central archive licensed under Creative Commons licenses.

The corpus contains about 100,000 concept annotations which point to seven ontologies/terminologies:

- Chemical entities of Biological Interest (ChEBI) (Degtyarenko et al., 2008)
- Cell Ontology[3]
- Entrez Gene (Maglott et al., 2005)
- Gene Ontology (biological process, cellular component, and molecular function) (Ashburner et al., 2000)
- the US National Center for Biotechnology Information (NCBI) Taxonomy[4]
- Protein Ontology[5]
- Sequence Ontology (Eilbeck et al., 2005)

Each of the 67 articles contains also linguistic information, such as tokenized sentences, part-of-speech information, parse trees, and dependency trees. Articles are represented in different formats, such as plain text or XML, and are easily navigable with common resources, such as the Knowtator plugin for the Protégé software.[6]

To make references to documents in the CRAFT corpus easily retrievable for the reader, when we

[1] The full CRAFT corpus comprises another 30 annotated articles, which are reserved for future competitions and have to date not been released.

[2] http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

[3] https://github.com/obophenotype/cell-ontology/

[4] http://www.ncbi.nlm.nih.gov/taxonomy

[5] http://pir.georgetown.edu/pirwww/index.shtml

[6] http://knowtator.sourceforge.net/

will refer to an article contained in the corpus we will list the name of its corresponding XML file as contained in the corpus distribution, its PubMed Central ID (PMCID), and its PubMed ID (PMID).[7]

3.2 OntoGene

The OntoGene group has developed an approach for biomedical entity recognition based on dictionary lookup and flexible matching. Their approach has been used in several competitive evaluations of biomedical text mining technologies, often obtaining top-ranked results (Rinaldi et al., 2008; Rinaldi et al., 2010; Rinaldi et al., 2012a; Rinaldi et al., 2014). Recently, the core parts of the pipeline have been implemented in a more efficient framework using Python (Colic, 2016). It offers a flexible interface for performing dictionary-based TTE.

OntoGene's term annotation pipeline accepts a range of input formats, e.g. PubMed Central full-text XML, gzipped chunks of Medline abstracts, BioC,[8] or simply plain text. It provides the annotated terms along with the corresponding identifiers either in a simple tab-separated text file, in brat's standoff format,[9] or – again – in BioC. It allows for easily plugging in additional components, such as alternative NLP preprocessing methods or postfiltering routines.

In the present work, the pipeline was configured as follows: After sentence splitting, the input documents were tokenized with a simple method based on character class: Any contiguous sequence of either alphabetical or numerical characters was considered a token, whereas any other characters (punctuation and whitespace) were considered token boundaries and were ignored during the dictionary look-up. This lossy tokenization already has a normalizing effect, in that it collapses spelling variants which arise from inconsistent use of punctuation symbols, e.g. "SRC 1" vs. "SRC-1" vs. "SRC1". (A similar approach is described by Verspoor et al. (2010), which refer to it as "regularization".) All tokens are then converted to lowercase, except for acronyms that collide with a word from general language (e.g. "WAS"). We enforced a case-sensitive match in these cases by using a list of the most frequent English words. As a further normalization step, Greek letters were expanded to their letter name in Latin spelling, e.g. $\alpha \rightarrow$ *alpha*, since this is a common alternation.

For term matching, we compiled a dictionary resource using the Bio Term Hub (Ellendorff et al., 2015). The Bio Term Hub is a large biomedical terminology resource automatically compiled from a number of curated terminology databases. Its advantage lies in the ease of access, in that it provides terms and identifiers from different sources in a uniform format. It is accessible through a web interface,[10] which recompiles the resource on request and provides it as a tab-separated text file.

Selecting the seven ontologies used in CRAFT resulted in a term dictionary with 20.2 million entries. Based on preliminary tests, we removed all entries with terms shorter than two characters or terms consisting of digits only; this reduced the number of entries by less than 0.3%. In the OntoGene system, the entries of the term dictionary were then preprocessed in the same way as the documents. Finally, the input documents were compared to the dictionary with an exact-match strategy.

3.3 Distiller

Distiller[11] is an open source framework written in Java and R for machine learning, introduced in Basaldella et al. (2015). While the framework has its roots in the work of Pudota et al. (2010), thus focusing on the task of automatic keyphrase extraction (herein AKE), Distiller's framework design allows us to adapt its pipeline to various purposes.

AKE is the problem of extracting *relevant* phrases from a document (Turney, 2000), and the difference with TTE is that, while the former is interested in a *small set* of *relevant* phrases from the source document, the latter is interested in *all domain-specific* terms.

While AKE can be performed using unsupervised techniques, the most successful results have been obtained using a supervised machine learning approach (Lopez and Romary, 2010). Supervised AKE is performed using a quite common pipeline: first, the candidate keyphrases are generated, using some kind of linguistic knowledge;

[7] We will not include articles from the CRAFT corpus in the references as they are *not* actual bibliography for the purposes of this work.

[8] http://bioc.sourceforge.net/

[9] `http://brat.nlplab.org/standoff.html`

[10] `http://pub.cl.uzh.ch/purl/biodb/`

[11] `https://github.com/ailab-uniud/distiller-CORE`

then, the AKE algorithm filters the candidates assigning them some features which are in turn used to train a machine learning algorithm, which is able to classify "correct" keyphrases. These keyphrases can be then used for several purposes, such as document indexing, filtering and recommendation (De Nart et al., 2013).

To adapt Distiller to perform TTE effectively, we substituted the candidate generation phase with the output of OntoGene, i.e. candidate technical terms become the potential "key phrases". This configuration is then evaluated as our baseline. Next, we gradually add new features into the system to train a machine learning model specialized in the actual TTE task, and assess the improvements in the performance of the system.

4 Features

4.1 Baseline

First, we evaluated the performance of the OntoGene/Distiller system using the same feature set used in the original keyphrase extraction model presented by Basaldella et al. (2015), which contains:

Frequency The frequency of the candidate in the document, also known as TF.

Height The relative position of the *first* appearance of the candidate in the document.

Depth The relative position of the *last* appearance of the candidate in the document.

Lifespan The distance between the first and the last appearance of the candidate.

TF-IDF The peculiarity of the candidate with respect to the current document and the CRAFT corpus. This is a very common feature both in the AKE and TTE fields.

Abstract Presence A flag set to 1 if the candidate appears in the abstract, 0 otherwise. This is motivated by the fact that often keyphrases are found to appear in the abstract.

This small feature set is the baseline of the experimental evaluation performed on the proposed approach.

4.2 Feature Set 1

To improve the performance of the TTE task we start to augment our feature set by introducing features that should be able to catch some more fine-grained information about the candidate terms.

Title Presence A flag which is set to 1 if the term appears in the title of the document and 0 otherwise, much like the *Abstract Presence* feature.

Symbols Count A counter for the number of punctuation symbols, i.e. not whitespaces and not alpha-numeric characters, appearing in the candidate term.

Uppercase Count A counter for the number of uppercase characters in the candidate term.

Lowercase Count A counter for the number of lowercase characters in the candidate term.

Digits Count A counter for the number of digits in the candidate term.

Space Count A counter for the number of spaces in the candidate term.

Greek Flag A flag that is set to 1 if the candidate contains a Greek letter in spelled-out form, like "alpha", "beta", and so on.

These features offer a good improvement for detecting the particular shape that a technical term could have. For example, from the document `PLoS Biol-2-1-314463.nxml` (PMC: PMC314463, PMID: 14737183) we have the term *"5-bromo-4-chloro-3-indolyl beta-D-galactoside"*. This term contains:

- A spelled-out Greek letter, *beta*;
- An uppercase letter;
- Seven symbols (dashes);
- A whitespace.

Without the new features this information would have been lost, so it *may* have been much harder to recognize the term as a technical one.

4.3 Feature Set 2

In this step we add even more features aimed at detecting more fine-grained information about candidate terms. The new features are:

Dash flag Dashes are one of the most (if not the most) common symbols found in technical terms. This flag is set to 1 if the term contains a dash, 0 otherwise.

Ending number flag This flag is set to 1 if the term ends with a number, 0 otherwise.

Inside capitalization This flag is set to 1 if the term contains an uppercase letter which is not at the beginning of a token.

All uppercase This flag is set to 1 if the term contains only uppercase letters, 0 otherwise.

All lowercase This flag is set to 1 if the term contains only lowercase letters, 0 otherwise.

4.4 Feature Set 3: Affixes

This feature set adds information about the affixes (i.e. prefixes and suffixes) of the words. This information is particularly useful in the biomedical field, since affixes in this field convey often a particular meaning: for example, words ending with "*ism*" are typically diseases, words starting with "*zoo*" refer to something from the animal life, and so on. Another example is the naming of chemical compounds: for example, many ionic compounds have the suffix "*ide*", such as *Sodium Chloride* (the common table salt).

Using the Bio Term Hub resource, we compiled a list of all the prefixes and suffixes of two or three letters from the following databases:

- Cellosaurus,[12] from the Swiss Institute of Bioinformatics;
- Chemical compounds found in the Toxicogenomics Database (CTD),[13] from the North Carolina State University Comparative;
- Diseases found in the CTD;
- EntrezGene (Maglott et al., 2005);
- Medical Subject Headings (MeSH),[14] from the US National Center for Biotechnology Information (restricted to the subtrees "organisms", "diseases", and "chemicals and drugs");
- Reviewed records from the Universal Protein Resource (Swiss-Prot),[15] developed by the UniProt consortium, which is a joint USA-EU-Switzerland project.

[12] http://web.expasy.org/cellosaurus/
[13] http://ctdbase.org/
[14] http://www.ncbi.nlm.nih.gov/mesh
[15] http://www.uniprot.org/

Since not all affixes are equally important, the affixes list needs to be cut at some point. While a trivial decision could have been to pick the top 100 or 10% ranked prefixes and suffixes, our choice was to let the machine learning algorithm decide by itself where to apply the cut.

To obtain this goal, each affix a from a database D is assigned a normalized score $s \in [0, 1]$ computed this way:

$$s(a) = \frac{\text{freq}(a, D)}{\max(\{\text{freq}(a_1, D) \dots \text{freq}(a_{|D|}, D)\})}$$

where $\text{freq}(a, D)$ is the frequency of an affix a in D. This way we obtain a simple yet effective mechanism to let a ML algorithm learn which of affixes are the most important.

It is also worth noting that since we generate scores for prefixes and affixes of two and three letters from six databases, we have a total of $2 \times 2 \times 6 = 24$ features generated with this approach.

4.5 Feature Set 4: Removing AKE Features

Now that we have many features that are more specific for the technical term extraction field, we remove the baseline feature set, which was tailored on keyphrase extraction, to use only the features aimed at recognizing technical terms.

These features (*depth*, *height*, *lifespan*, *frequency*, *abstract presence*, *title presence*, *TF-IDF*) are specific for the AKE field and supposedly bring little value on knowing if a term is technical or not. In fact, a term may appear just once in a random position of the text, and still be technical; the same does not hold for a keyphrase, which is assumed to appear many times in specific positions (introduction, conclusions...) in the text.

4.6 Test Hardware

Both OntoGene and Distiller have been tested on a laptop computer with an Intel i7 4720HQ processor running at 2,6GHz, 16 GB RAM and a Crucial M.2 M550 SSD. The operating system was Ubuntu 15.10.

The speed was of 16275 words/second for OntoGene and 4745 words/second for Distiller. OntoGene requires an additional time of about 25 second to load the dictionary at start up, but since this operation is run only once we do not consider it for the average.

Metric	OntoGene	Baseline	FS1	FS2	FS3	FS4
Precision	0.342	0.692	0.682	0.710	0.771	**0.853**
Recall	0.550	0.187	0.247	0.264	0.325	**0.368**
F1-Score	0.421	0.294	0.362	0.385	0.457	**0.515**

Table 1: Scores obtained with the Distiller/Ontogene pipeline using a MLP trained on the CRAFT corpus. In the column headers, "FSn" stands for "Feature Set n".

System	Precision	Recall	F1
MMTx	0.43	0.40	0.42
MGrep	0.48	0.12	0.19
Concept Mapper	0.48	0.34	0.40
cTakes Dictionary Lookup	0.51	0.43	0.47
cTakes Fast Lookup	0.41	0.4	0.41
NOBLE Coder	0.44	0.43	0.43
OntoGene	0.34	**0.55**	0.42
OntoGene+Distiller	**0.85**	0.37	**0.51**

Table 2: Comparison of the scores obtained with OntoGene, with the combined OntoGene/Distiller pipeline and the scores obtained in Tseytlin et al. (2016).

5 Results

Using the feature sets defined above, we trained a neural network to classify technical terms. The network used is a simple multi-layer perceptron, with one hidden layer containing twice the number of neurons of the input layer and configured to use maximum conditional likelihood. The network is trained using 47 documents of the CRAFT corpus as training set and its performance is evaluated on the remaining 20, which in turn form the test set.

We also experimented using a C5.0 decision tree, but with unsatisfactory results (the performance decreases with the number of features) so we do not include its analysis in this paper.

The metrics used are simple Precision, Recall and F1-Score. Table 1 presents the performance of the different iterations of the proposed system. Plain OntoGene obtains 55.0% recall and 34.2% precision, while the baseline AKE feature set improves the precision score with 69.2% score in precision but shows a dramatic drop in recall to 18.7%.

It can be seen that the introduction of TTE-specific features brings an important improvement in recall, with a 6% improvement between the baseline and Feature Set 1. Together with a small drop in precision by 1%, it augments the F1-score by 7 points.

Feature Set 2 performs slightly better than Feature Set 1, with a general improvement between 2% and 3%. Then Feature Set 3, adding the affixes, brings a great improvement of 7% F1-Score, thanks to a general improvement of precision and recall of the same order.

Finally, it is clear that the Feature Set 4 (i.e. all the TTE-focused features, without the AKE-focused ones) is the best performing one. The obtained precision of 85.3% is a large improvement from the baseline of 69% and more than twice the precision of the raw OntoGene output, which is just 34.2%.

More importantly, recall rises from 18.7% to 36.8% (over a theoretical maximum of 55.0% of the raw OntoGene output). Feature Sets 3 and 4 also obtain a better F1-score than OntoGene, with 45.7% and 51.5%, respectively, while the score obtained by the OntoGene system is just 42.1%.

To compare our pipeline with similar TTE/CR software, we use the results by Tseytlin et al. (2016), which compared the NOBLE coder with MMTx,[16] Concept Mapper,[17] cTAKES[18] and MGrep (Dai et al., 2008), as shown in Table 2. We can see that our result outperforms the 0.47 F1-score obtained by the best performing system, i.e. cTAKES Dictionary Lookup, in that compar-

[16] https://mmtx.nlm.nih.gov/MMTx/

[17] https://uima.apache.org/sandbox.html#concept.mapper.annotator

[18] http://ctakes.apache.org/

ison. This result is achieved thanks to the high precision obtained by Distiller's machine learning stage, which boosts precision to 78%, while the precision of the best performing system in the same comparison is just 51%.

We must stress that our results are not directly comparable to the ones in Tseytlin et al. (2016), for three reasons. Firstly, we evaluate the combined pipeline only on a portion of the dataset, since a training set is needed for the Distiller system. Secondly, we do not do concept disambiguation, but rather we consider a true positive whenever our pipeline marks a term that spans the same text region as a CRAFT annotation, regardless of what entity is associated with this term, which is an easier task than concept recognition. On the other hand, Tseytlin et al. (2016) count also *partial positives*, i.e. if the software annotation does not exactly overlap with the gold annotation, they allocate *one-half match* in both precision and recall. Instead, while evaluating our system, we count only exact matches, giving a disadvantage to our system.

Still, the more than doubling of precision from the dictionary-only approach is noteworthy, especially because it compensates the loss in recall well enough to have a general improvement in F1-score. The comparison, while not completely fair, shows that the high precision of our system is hardly matched by other approaches.

The biggest drawback of our approach is the relatively low recall still obtained by the OntoGene pipeline, which puts an upper bound to the recall obtainable by the complete pipeline. The 55% recall score obtained on the CRAFT corpus is not a bad result *per se*, as it is better to the best performance obtained in Tseytlin et al. (2016) by NOBLE Coder and cTAKES Dictionary Lookup. Nevertheless, we believe that recall can be improved by addressing some specific issues we analyze in greater detail in Section 6.2.

6 Error Analysis

6.1 False Positives and CRAFT Problems

Looking at the errors performed by our system, we believe that some outcomes that seem to be false positive should actually be marked as true positives. Take as an example document `PLoS Genet-1-6-1342629.nxml` (PMC-ID: PMC1315279, PMID: 16362077). In the Discussion section, we have (emphasis ours):

> Serum levels of **estrogen** decreased in aging Sam68$^{-/-}$ females as expected; however, the leptin levels decreased in aged Sam68$^{-/-}$ females.

The term *estrogen* is not annotated in the CRAFT corpus, even though it is found in the ChEBI resource. OntoGene, on the other hand, recognizes this as a relevant term. The same holds for the two other occurrences of this term in the same article.

In the Results section of the same document, we have

> Given the apparent enhancement of mineralized nodule formation by Sam68$^{-/-}$ bone marrow stromal cells ex vivo and the phenotype observed with **short hairpin RNA** (**shRNA**)-treated C3H10T1/2, we stained sections of bone from 4- and 12-month-old mice for evidence of changes in marrow adiposity.

Here, OntoGene annotates the Sequence-Ontology term *shRNA* both in its full and abbreviated form. Nevertheless, they are missing from the CRAFT annotations (along with 6 more occurrences of *shRNA*); however, CRAFT provides annotations to parts of the term (*hairpin* and *RNA*).

Then, in the Materials and Methods section, we have

> Briefly, cells were plated on glass coverslips or on **dentin** slices in 24-well cluster plates for assessment of cell number and pit number, respectively.

Again, the term *dentin*, which is present in the Cell Ontology, is found by OntoGene but absent from the CRAFT corpus, together with 5 more occurrences of the same term.

Looking at this example document, we can see that the annotation of the CRAFT corpus seems to be somewhat inconsistent. While the reasons may be various and perfectly reasonable (e.g. the guidelines might explicitly exclude the mentioned terms in that context), this fact may affect the training and evaluation of our system.

6.2 Causes of Low Recall

Many terms annotated in the CRAFT corpus are missed by the OntoGene pipeline. As a general observation, the OntoGene pipeline – originally geared towards matching gene and protein names

– is not optimally adapted to the broad range of term types to be annotated. A small number of the misses (less than 1%) is caused by the enforced case-sensitive match for words from the general vocabulary (such as "Animal" at the beginning of a sentence). Another portion (around 5%) are due to the matching strategy, in that the aggressive tokenization method removed relevant information, such as trailing punctuation symbols or terms consisting entirely of punctuation (e.g. "+"). Approximately 9% are short terms of one or two characters' length, which had been excluded from the dictionary a priori, as described above. A major portion, though, are inflectional and derviational variants, such as plural forms or derived adjectives (e.g. missed "mammalian" besides matched "mammal"). Some CRAFT annotations include modifiers that are missing from the dictionary, e.g. the protein name "TACC1" is matched on its own, but not when disambiguated with a species modifier such as "mouse TACC1"/"human TACC1". Other occasional misses include paraphrase ("piece of sequence") or spelling errors ("phophatase" instead of "phosphatase").

7 Conclusions and Future Work

In this paper we have presented and evaluated an approach towards efficient recognition of biomedical entities in the scientific literature. Although some limitations are still present in our system, we believe that this approach has the potential to deliver high quality entity recognition, not only for the scientific literature, but on any related form of textual document. We have analyzed the limitations of our approach, clearly discussing the causes of the low recall when evaluated over the CRAFT corpus. The results show that the post-annotation filtering step can significantly increase precision at the cost of a small loss of recall. Additionally, the approach provides a good ranking of the candidate entities, thus enabling a manual selection of the best terms in the context of an assisted curation environment.

As for future work, we intend to improve coverage of the OntoGene pipeline with respect to the CRAFT annotations. Based on the false-negative analysis, the next steps include: (1) use a stemmer or lemmatizer, (2) optimize the punctuation handling, (3) revise the case-sensitive strategy.

We also plan to improve Distiller's machine learning phase, adding more features to the neural network classifier or switching to other approaches used in literature, such as conditional random fields (Leaman et al., 2015). Another approach that we will investigate is to make the algorithm able to disambiguate between different term types proposed by the OntoGene pipeline, using a multi-class classifier.

References

Saber A Akhondi, Ewoud Pons, Zubair Afzal, Herman van Haagen, Benedikt FH Becker, Kristina M Hettne, Erik M van Mulligen, and Jan A Kors. 2016. Chemical entity recognition in patents by combining dictionary-based and statistical approaches. *Database*, 2016:baw061.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. 2000. Gene Ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29.

Sophie Aubin and Thierry Hamon. 2006. Improving term extraction with terminological resources. In *Advances in Natural Language Processing*, pages 380–387. Springer.

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, Judith A Blake, et al. 2012. Concept annotation in the CRAFT corpus. *BMC bioinformatics*, 13(1):1.

Marco Basaldella, Dario De Nart, and Carlo Tasso. 2015. Introducing Distiller: a unifying framework for knowledge extraction. In *Proceedings of 1st AI*IA Workshop on Intelligent Techniques At Libraries and Archives co-located with XIV Conference of the Italian Association for Artificial Intelligence (AI*IA 2015)*. Associazione Italiana per l'Intelligenza Artificiale.

David Campos, Sérgio Matos, and José Luís Oliveira. 2013. A modular framework for biomedical concept recognition. *BMC Bioinformatics*, 14:281.

Nicola Colic. 2016. Dependency parsing for relation extraction in biomedical literature. Master's thesis, University of Zurich, Switzerland.

Manhong Dai, Nigam H Shah, Wei Xuan, Mark A Musen, Stanley J Watson, Brian D Athey, Fan Meng, et al. 2008. An efficient solution for mapping free text to ontology terms. *AMIA Summit on Translational Bioinformatics*, 21.

Dario De Nart, Felice Ferrara, and Carlo Tasso. 2013. Personalized access to scientific publications: from recommendation to explanation. In *User Modeling, Adaptation, and Personalization*, pages 296–301. Springer Berlin Heidelberg.

Kirill Degtyarenko, Paula De Matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan McNaught, Rafael Alcántara, Michael Darsow, Mickaël Guedj, and Michael Ashburner. 2008. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic acids research*, 36(suppl 1):D344–D350.

Karen Eilbeck, Suzanna E Lewis, Christopher J Mungall, Mark Yandell, Lincoln Stein, Richard Durbin, and Michael Ashburner. 2005. The Sequence Ontology: a tool for the unification of genome annotations. *Genome biology*, 6(5):R44.

Tilia Renate Ellendorff, Adrian van der Lek, Lenz Furrer, and Fabio Rinaldi. 2015. A combined resource of biomedical terminology and its statistics. In Thierry Poibeau and Pamela Faber, editors, *Proceedings of the 11th International Conference on Terminology and Artificial Intelligence*, pages 39–49, Granada, Spain.

Ken-ichiro Fukuda, Tatsuhiko Tsunoda, Ayuchi Tamura, Toshihisa Takagi, et al. 1998. Toward information extraction: identifying protein names from biological papers. In *Pac Symp Biocomput*, pages 707–718.

Christopher Funk, William Baumgartner, Benjamin Garcia, Christophe Roeder, Michael Bada, K Bretonnel Cohen, Lawrence E Hunter, and Karin Verspoor. 2014. Large-scale biomedical concept recognition: an evaluation of current automatic annotators and their parameters. *BMC bioinformatics*, 15(1):1.

Tudor Groza and Karin Verspoor. 2015. Assessing the impact of case sensitivity and term information gain on biomedical concept recognition. *PloS one*, 10(3):e0119091.

Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. 2015. tmChem: a high performance approach for chemical named entity recognition and normalization. *J. Cheminformatics*, 7(S-1):S3.

Patrice Lopez and Laurent Romary. 2010. HUMB: automatic key term extraction from scientific articles in GROBID. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 248–251. Association for Computational Linguistics.

Donna Maglott, Jim Ostell, Kim D Pruitt, and Tatiana Tatusova. 2005. Entrez Gene: gene-centered information at NCBI. *Nucleic acids research*, 33(suppl 1):D54–D58.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Nirmala Pudota, Antonina Dattolo, Andrea Baruzzo, Felice Ferrara, and Carlo Tasso. 2010. Automatic keyphrase extraction and ontology mining for content-based tag recommendation. *International Journal of Intelligent Systems*, 25(12):1158–1186.

Fabio Rinaldi, Thomas Kappeler, Kaarel Kaljurand, Gerold Schneider, Manfred Klenner, Simon Clematide, Michael Hess, Jean-Marc von Allmen, Pierre Parisot, Martin Romacker, and Therese Vachon. 2008. OntoGene in BioCreative II. *Genome Biology*, 9(Suppl 2):S13.

Fabio Rinaldi, Gerold Schneider, Kaarel Kaljurand, Simon Clematide, Therese Vachon, and Martin Romacker. 2010. OntoGene in BioCreative II.5. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):472–480.

Fabio Rinaldi, Simon Clematide, and Simon Hafner. 2012a. Ranking of CTD articles and interactions using the OntoGene pipeline. In *Proceedings of the 2012 BioCreative workshop*, Washington D.C., April.

Fabio Rinaldi, Gerold Schneider, Simon Clematide, and Gintare Grigonyte. 2012b. Notes about the OntoGene pipeline. In *AAAI-2012 Fall Symposium on Information Retrieval and Knowledge Discovery in Biomedical Text, November 2-4, Arlington, Virginia, USA*.

Fabio Rinaldi, Simon Clematide, Hernani Marques, Tilia Ellendorff, Raul Rodriguez-Esteban, and Martin Romacker. 2014. OntoGene web services for biomedical text mining. *BMC Bioinformatics*, 15(Suppl 14):S6.

Fabio Rinaldi. 2012. The OntoGene system: an advanced information extraction application for biological literature. *EMBnet.journal*, 18(Suppl B):47–49.

Yutaka Sasaki, Yoshimasa Tsuruoka, John McNaught, and Sophia Ananiadou. 2008. How to make the most of NE dictionaries in statistical NER. *BMC bioinformatics*, 9(11):1.

Damian Szklarczyk, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi P. Tsafou, Michael Kuhn, Peer Bork, Lars J. Jensen, and Christian von Mering. 2015. STRING v10: protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Research*, 43(D1):D447–D452.

Eugene Tseytlin, Kevin Mitchell, Elizabeth Legowski, Julia Corrigan, Girish Chavan, and Rebecca S Jacobson. 2016. NOBLE – Flexible concept recognition for large-scale biomedical natural language processing. *BMC bioinformatics*, 17(1):1.

Peter D Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336.

Karin Verspoor, Christophe Roeder, Helen L Johnson, Kevin Bretonnel Cohen, William A Baumgartner Jr, and Lawrence E Hunter. 2010. Exploring species-based strategies for gene normalization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 7(3):462–471.