Journal of Cheminformatics

RESEARCH ARTICLE

Open Access

OGER++: hybrid multi-type entity recognition

Lenz Furrer[1†], Anna Jancso[1†], Nicola Colic[1] and Fabio Rinaldi[1,2*]

Abstract

Background: We present a text-mining tool for recognizing biomedical entities in scientific literature. OGER++ is a hybrid system for named entity recognition and concept recognition (linking), which combines a dictionary-based annotator with a corpus-based disambiguation component. The annotator uses an efficient look-up strategy combined with a normalization method for matching spelling variants. The disambiguation classifier is implemented as a feed-forward neural network which acts as a postfilter to the previous step.

Results: We evaluated the system in terms of processing speed and annotation quality. In the speed benchmarks, the OGER++ web service processes 9.7 abstracts or 0.9 full-text documents per second. On the CRAFT corpus, we achieved 71.4% and 56.7% F1 for named entity recognition and concept recognition, respectively.

Conclusions: Combining knowledge-based and data-driven components allows creating a system with competitive performance in biomedical text mining.

Keywords: Named entity recognition, Concept recognition, Natural language processing, Machine learning

Background

Text mining is often the only answer to retrieving specific information in the vast amount of biomedical scientific literature. Reliably extracting basic entities such as chemicals, genes/proteins, diseases, or organisms is the foundation of most approaches to text mining. The task of detecting spans of text that denote an entity of interest is usually referred to as *named entity recognition* (NER). It is commonly modeled as a tagging problem, where the text is a sequence of tokens that are classified as relevant or irrelevant and if multiple entity types are targeted assigned a type. In the closely related task of *concept recognition* (CR, often also referred to as *entity linking*, *normalisation*, or *grounding*), entities are additionally annotated with unique identifiers.

In terms of methodology, many approaches have been taken towards biomedical entity recognition. The evolution of methods reflects the advances that can be observed in all areas of natural language processing (NLP). Early systems were based on hand-written rules for extracting entities [1–3]. Over the last decade, supervised machine-learning systems have become very popular. For NER, Conditional Random Fields (CRF) have long dominated the field [4–7]. Knowledge-based approaches using hand-crafted resources like gazetteers are widespread among CR systems [8–11], even though data-driven components are used frequently to generate and/or rank entity candidates [12–16]. In multi-model architectures, multiple models are combined in a serial [17] or parallel manner (ensemble systems) [18–20] or use unlabeled data for improving domain representation [21]. Approaches to tackling both NER and CR include sequential pipelines [22] and joint models [23, 24]. Very recently, the renaissance of neural networks (NN) observable in many subfields of artificial intelligence and NLP finally found its way to NER for the biomedical domain [25–29] and even CR [30].

In this work, we present OGER++, a hybrid NER-CR system for text mining in the biomedical domain. It combines a fast, dictionary-based entity recognizer and normalizer with a corpus-based disambiguation filter. The technical and qualitative performance of previous

*Correspondence: fabio.rinaldi@uzh.ch
†Lenz Furrer and Anna Jancso equal contributor
[1] Institute of Computational Linguistics, University of Zurich, Andreasstr. 15, 8050 Zürich, Switzerland
Full list of author information is available at the end of the article

versions were described in [31, 32], respectively. More recently, we wrote an application based on OGER for the OpenMinTeD platform [33]. In the present work, we describe new experimental results and benchmarks performed with the current version of the software. The code base of OGER is freely available from https://github.com/OntoGene/OGER; the demo web service is running at https://pub.cl.uzh.ch/purl/OGER.

Methods

OGER—OntoGene's Entity Recognizer—is a versatile, extensible software package written for multiple applications. It can be used as a Python library, executed as a command-line tool, or run as a REST server with an API and a browser interface. Through a software hook, it can be extended by the user through custom Python modules. By OGER++, we refer to a publicly accessible web service hosted at our institute which provides on-the-fly document annotation using a large terminology resource and corpus-based disambiguation. In the following, we describe this actively running instance of OGER.

OGER++ performs document annotation in four steps: (1) document structure parsing, (2) entity recognition/normalization, (3) disambiguation, (4) serialization. A wide range of input and output formats are supported (Steps 1 and 4), including plain-text, PubMed/PMC XML, BioC [34] (XML and JSON), and PubAnnotator JSON [35, 36], among others. In the format conversion, textual content and basic structure (sections) are retained, as well as a limited set of metadata (document ID for most formats, all metadata for BioC). In Step 2, a dictionary-based strategy is applied to locate mentions of biomedical entities in the text and link them to identifiers of curated terminology resources, as described in the next section. This procedure frequently generates ambiguous annotations (i.e. the same text span is linked to multiple entities), which are addressed in Step 3, as discussed in the subsequent section.

Dictionary-based entity recognition and normalization

OGER has its roots in the OntoGene term annotation pipeline, a knowledge-based information extraction system for scientific biomedical literature that has been successfully applied to a range of entity types (genes/proteins, chemicals, diseases, among others [37–40]). It has been reimplemented from scratch in Python and is being developed continuously. As such, it has seen considerable improvements in terms of flexibility and processing speed.

The core recognition procedure relies on a list of target terms (the dictionary), which are connected to entity identifiers. Since extraction with an exact-match strategy would lead to very low coverage, we perform a series of preprocessing steps which have a normalizing effect. For example, the text is tokenized in an aggressive, lossy way which collapses spelling alternations like e.g. "SRC1"/"SRC 1"/"SRC-1" into a single representation. A more detailed description of the preprocessing steps can be found in [41].

At indexing time, each term (name) from the dictionary is converted to a sequence of tokens through the same preprocessing steps that are used for the documents (see Fig. 1 for an example), thus assuring that all potential matchings will be preserved. These token sequences are indexed in a hash table, which maps the term to its dictionary entry (containing the identifier and other metadata). In case of ambiguity (multiple entries have the same token sequence), the value of the hash table will contain multiple entries; for synonyms (multiple terms for the same concept), multiple entries are indexed. For an efficient look-up of variable-length sequences, an additional hash table maps the first token

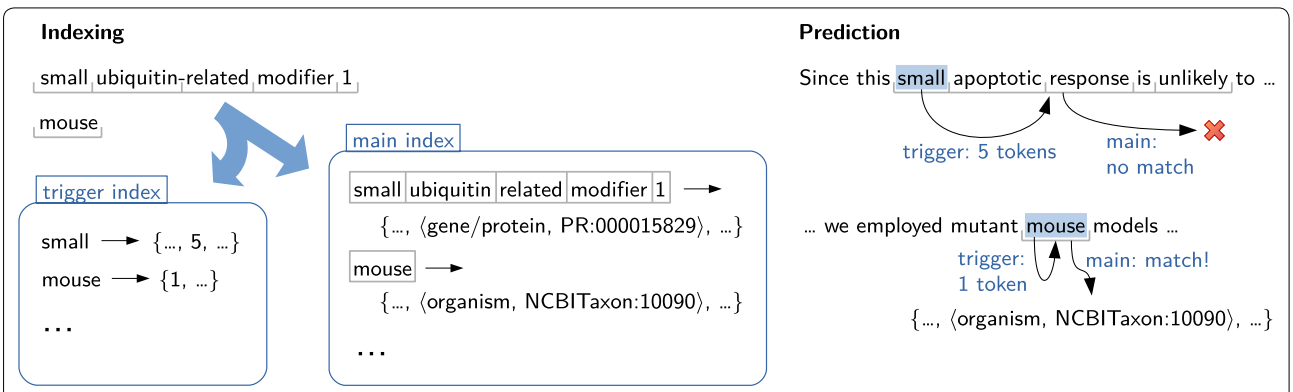

Fig. 1 Term indexing using two hash tables. The examples illustrate how dictionary entries are indexed (left) and how the look-up is performed (right)

of a term (trigger) to the length of the token sequence. At prediction time, each token of the text (preprocessed the same way as the dictionary terms) is looked up in the trigger index. If a match is encountered, candidate token sequences of appropriate length are extracted from the text, starting from the matching token. The extracted sequences are then looked up in the main index. Thanks to the trigger index, the number of look-ups per token is 1 in the common case (no trigger), i.e. complexity class $O(s)$ (best case) with respect to the number of tokens per sentence. Using only the main index, a look-up would be required for each contiguous subsequence of the sentence, i.e. $O(s^2)$ or, if the token count of the longest entity is known, $O(s \times t_{\max})$.

For the present work, we used two different configurations of terminology resources. In the experiment for evaluating annotation quality, we used the ontologies included in the CRAFT corpus [42], i.e. ChEBI [43], Cell Ontology [44], Gene Ontology [45], NCBI Taxonomy [46], Protein Ontology [47], and Sequence Ontology [48]. For the speed benchmarks, we used the default configuration of OGER's web service, which uses up-to-date versions of the resources mentioned above and, in addition, Cellosaurus [49], CTD chemicals and diseases [50], MeSH [51], Swiss-Prot [52], and Uberon [53]. All resources were aggregated and converted to a unified format using the Bio Term Hub, a meta-resource for collecting and combining curated terminology resources [54].

Corpus-based disambiguation

The dictionary-based concept-recognition module produces many spurious annotations. Words from the common vocabulary may be erroneously annotated as a biomedical entity (such as lead), and some terms are linked to identifiers of the wrong entity type (this often happens with abbreviations). Since OGER can produce multiple annotations for the same text span, the list of annotations might contain both correct and wrong results. Therefore, we augmented OGER with a postfilter component that removes spurious annotations.

The disambiguation procedure is illustrated in Fig. 2. For each annotated text span, the postfilter predicts a probability distribution over all entity types, including a label for not an entity. In the experiment with the CRAFT corpus (where a single text span can have multiple annotations), we applied the following heuristic to produce a label:

1. consider the highest-ranked entity type;
2. if the score difference between the two top-ranked types is less than a fixed threshold θ, consider the second-ranked entity type as well;
3. remove occurrences of not an entity from the list of labels to be considered.

The threshold θ was empirically set to 0.3 based on hyperparameter optimization with 5-fold cross-validation on the training set. This heuristic produces zero, one, or two labels per text span, which are not necessarily a subset of the annotations originally generated by OGER. Depending on the task, they are used differently: In the case of NER, the produced labels are emitted directly. This means that an annotation might be re-classified, i.e. given an entity type that was not among OGER's annotations. For the CR task, however, the concept identifiers are needed, therefore the original OGER annotations are used, restricted to the entries that match the postfilter's output. This means that any re-classified annotation is lost in CR, since no identifier can be provided.

The postfilter module is a machine-learning-based classifier that has to be trained on an annotated corpus. In the present work, we used the CRAFT corpus [42], which

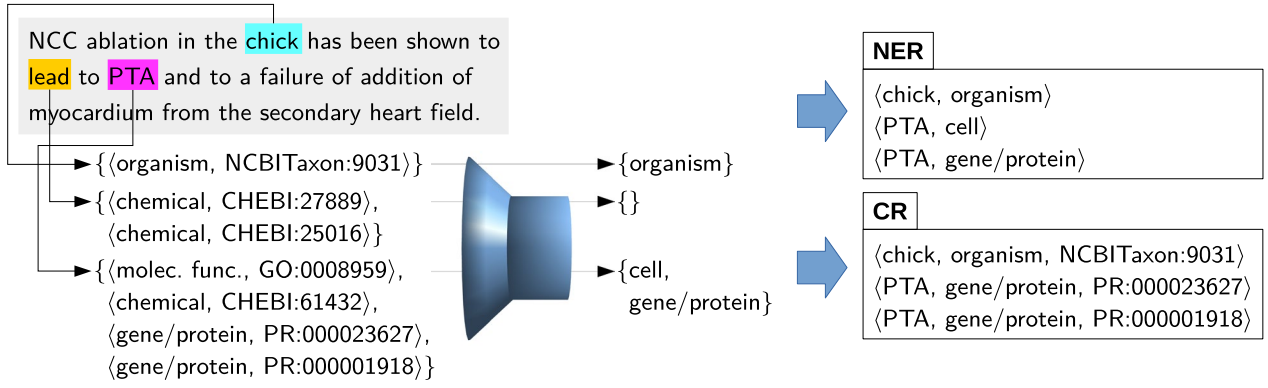

Fig. 2 Example illustrating the disambiguation procedure. The corpus-based postfilter accepts, rejects, or reclassifies annotations from the upstream concept-recognition module

is a collection of 67 full-text articles manually annotated for multiple entity types. The annotations cover chemicals, cell types, cellular components, organisms, genes/proteins, sequence features and the non-physical types biological processes and molecular functions. For our experiments, we excluded gene annotations linked to NCBI Gene (Entrez Gene) and conflated biological processes and molecular functions into a shared type BPMF. Annotations consisting of textually separated components were split into multiple, contiguous annotations. We divided the corpus into 47 documents for training and 20 for testing, using the same split as in our previous work [32].

The postfilter is implemented as a feed-forward neural network (NN). The initial design [32] was revised later [55] and integrated into OGER++. The key differences between the first and the current system are described in the following.

Firstly, both feature extraction and training of the NN is now performed in Python, thereby making it seamlessly work with the knowledge-based system implemented in the same programming language. The former system relied on a Java framework specialized on key-phrase extraction, plus a specialized learning module in R, to accomplish these tasks, thus making it very cumbersome to use in a pipeline. Secondly, a larger set of features was included as input to the NN. All thirteen features from the previous work were re-implemented. Four additional features were devised and evaluated:

- The **vowel:consonant** feature computes the proportion of vowels and consonants. Lower vowel counts are typical for certain entity types such as proteins.
- The **common vocabulary** feature computes whether the n-gram occurs in a common-language dictionary such as Hunspell [56]. Biomedical entities are less likely to appear in a common dictionary as can be seen in Fig. 3. Thus, this feature can help in deciding whether an n-gram should be ruled out as a biomedical entity mention. As Hunspell is intended to be used on single words, the percentages of terms known to Hunspell were calculated in two ways: In the "break-up" setting, the words of a term are looked up individually, while in the "no break-up" setting, they are passed to Hunspell as a whole. In the latter case, Hunspell always returns multi-word terms as not occurring in the dictionary. For some entity types, there are marked differences in the two percentages, notably for cells, biological processes, cellular components, sequences and organ/tissue. This means that terms of these entity types are frequently made up of common words. The current system performs no break-up of term as a split-up does not improve the accuracy of annotation.
- The **stop-words** feature computes whether the n-gram is a stop-word. Some stop-words also have a biomedical meaning and therefore appear in terminology resources. The NN can give lower weights to these words to decrease the rate of false-positives produced by these words. We used NLTKs [57] English stop-word list, which comprises 153 words.
- The **word embeddings** feature fetches the word embedding of an n-gram. Word embeddings add distributional knowledge for a given word. In our model, we used the pre-trained embeddings of [58], which target biomedical applications. For multi-word terms, which have no embedding, we used to take the word embedding of the head token, using the last token as an approximation which typically conveys the main meaning. The current system, however, performs an individual look-up for every token in the term and averages their respective embeddings using the mean to produce a single vector. We found that this improved the F1-scores for NER and CR by 0.3–0.4%, compared to using the word embedding of the head token.

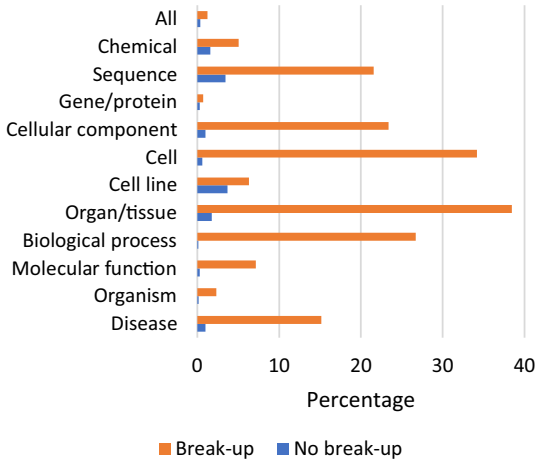

Fig. 3 Percentage of terms occurring in Hunspell

Experiments have shown that word embeddings are the most salient feature. In fact, using only word embeddings and excluding all other features only produced a small drop of 1 to 2% in the F1-score on the CRAFT corpus. This suggests that the influence of the other features is not very pronounced and that they might be redundant in future work. The public OGER web service uses three features only (common dictionary, stop-words, word embeddings).

A third main difference is that the previous system [32] trained separate NNs for each entity type, where a single output neuron makes a basic accept/reject decision given some threshold value. Our new system, however, trains a joint model by constructing a softmax output layer that computes a probability distribution over all entity types, as shown in Fig. 4. This has the advantage that the probabilities of different entity types become comparable and that only one model has to be loaded for predictions.

To give the NN filter capabilities, an additional output neuron for the label "not an entity" was added. For training, we used the rest of the words from the CRAFT corpus that were not explicitly annotated as biomedical in order for the NN to learn how common words look like. Note that the NN only receives single words as input in the case of common words, while in the case of biomedical entities, it can receive multi-word examples. The downside of this strategy is that the NN does not learn to remove irrelevant multi-word matches produced by the up-stream annotator.

To allow for multiple classifications of the same n-gram, as is the case for some biomedical datasets (e.g. the CRAFT corpus), entity types with the second-highest probability are also considered by defining a maximum probability difference to the most probable entity type.

Server architecture

An overview of the server architecture is given in Fig. 5. Incoming requests are expected to either include a PubMed or PMC ID (fetch command), or to contain an entire document in the request payload (upload command). In the case of a fetch request, the service fetches the referenced document using NCBI's efetch API [59]. The client can specify a number of parameters through the URL and an optional query string, such as the document input and output formats or the selection of terminologies to use for annotation. Different terminologies are maintained in separate instances of the dictionary-based annotation component as described above, called annotators. New annotators can be created by the client through another request (dict command, not shown in the figure); the Bio Term Hub makes use of this features to allow users to send newly compiled terminology resources to OGER. After annotation, the documents are passed to the postfilter for disambiguation and serialized into the requested output format, before being returned to the client.

Results and discussion

We assessed OGER++ with benchmarks for processing speed, an analysis of entity-type ambiguity, and an evaluation of annotation quality, as is discussed in the following sections.

Processing speed

The *technical interoperability and performance of annotation servers* (TIPS) task of the BioCreative V.5 challenge was a shared task designed to evaluate the efficiency and reliability of annotation servers in the biomedical domain. Among the participating systems, OGER was the fastest system (best results for *average response time* and *mean time per document volume*, team 122 in [60]). Additionally, we recently performed a series of benchmarks for measuring the processing

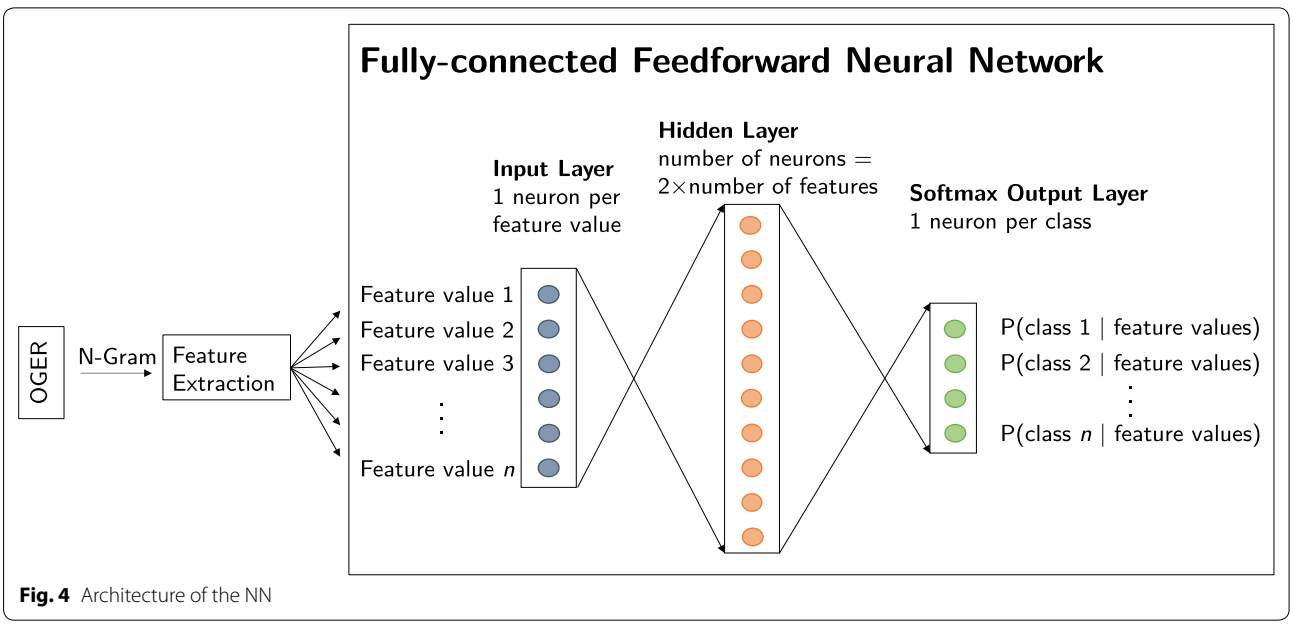

Fig. 4 Architecture of the NN

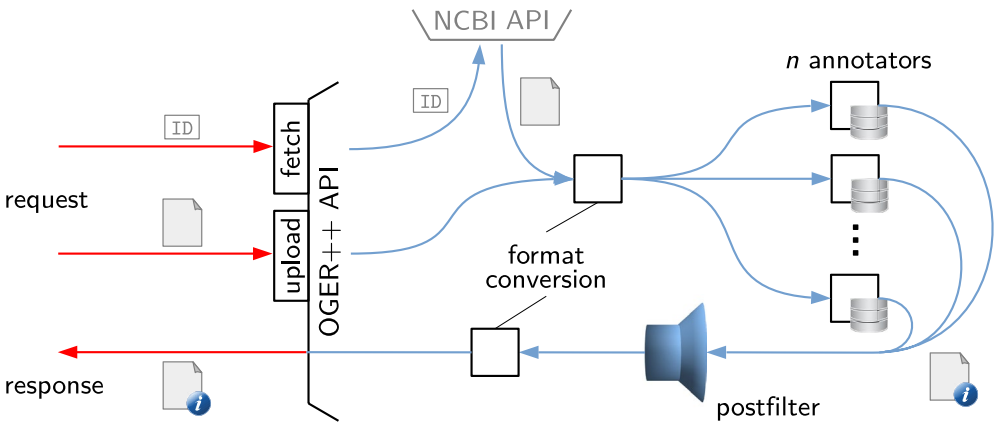

Fig. 5 System architecture of the OGER++ server

speed of OGER++. The results are summarized in Table 1. We analyzed two different document sizes (abstracts vs. full-text) and two different input formats (plain-text vs. NCBI XML). The same random sample of PubMed abstracts and PMC full-text documents was used for the different input formats.

The benchmarks were carried out using the public OGER web API. This web service is hosted on a virtual machine with 16 shared CPU cores and 128 G exclusive RAM. Each document was processed with a separate HTTP request in a serial fashion (no parallelization). Due to the requests being sent from the same physical machine on which the OGER service is run, network latency is expected to have negligible effect on the measurements; therefore, these results are not comparable to the average response time measured in the TIPS task (1.1 s per abstract, i.e. 10 times slower), where three separate HTTP requests between distant servers were necessary for each document. However, the current figures include the overhead required by the HTTP protocol. During the time of the tests, the server did not have a heavy load; in busy times, the processing times can be up to three times higher, even though OGER's service machine is prioritized by default.

Most time is spent in disambiguation, i.e. the NN predicting probabilities for each annotation. This can be clearly seen by comparing to the last line in the table, where full-text documents were processed without disambiguation, which leads to 20 times faster processing on average. Document size affects processing time greatly, as abstracts are processed more than 10 times faster than full-text documents. This is best explained by the higher number of annotated terms in longer texts. The input format has only a marginal effect both on processing time and the number of annotations the absence of structural mark-up tends to accelerate processing and has an influence on term matching.

Entity-type ambiguity

In order to estimate the degree of ambiguity in a multi-type entity-recognition setting, we performed an experiment using OGER without its disambiguation module. Using a large dictionary with 5.6 million names for a total 2.9 million concepts of 11 different entity types, we automatically annotated a random sample of 3 million PubMed abstracts. Since disambiguation was disabled, each annotated mention was tagged with one or more entity types. We used these data to compute a confusion matrix

Table 1 Average processing time analysis for different document formats and sizes

Size	Format	Documents	doc/s	kiB/s	ann/s	kiB/s (macro)	ann/s (macro)	ann/doc
Abstracts	txt	1000	9.73	8.27	462.96	11.75	239.70	47.56
Abstracts	xml	1000	9.45	57.26	449.43	222.44	241.55	47.56
Full-text	txt	529	0.89	16.97	866.89	18.95	621.95	979.09
Full-text	xml	529	0.88	47.44	862.01	64.16	620.00	979.37
Full-text (no disambiguation)	txt	529	17.82	341.64	28072.22	350.24	18569.08	1575.27

For kiB/s and ann/s, micro- and macro-average are given separately

of names that are shared among different entity types, measured by their occurrence in the scientific literature. When comparing dictionary entries in their exact spelling, there is almost no overlap across entity types; however, the relaxed matching scheme used for annotation introduces a significant number of collisions, as can be seen in Fig. 6. Please note that the true type is unknown in this setting, and that a considerable fraction of annotations is potentially spurious, i.e. words of common language that are erroneously annotated as a biomedical entity. However, these figures give a realistic estimate of how hard the task of the disambiguation module is.

CRAFT evaluation

We performed an evaluation on 20 articles from the CRAFT corpus using the metrics precision, recall and F1-score. We evaluated the correctness of the system output at two different levels: entity type (NER evaluation) and identifier (CR evaluation), as is described in the following sections.

NER evaluation

In the NER-level evaluation, we considered a prediction to be correct (true positive) if it matched the span (character offsets) and entity type of a ground-truth annotation. We required the span to match exactly, i.e. no credit was given for predictions that partially overlapped with a true annotation. Table 2 shows micro-averaged precision, recall and F1-scores broken down by entity type for three different systems: the knowledge-based system (OG), the previous hybrid system (OG + Dist) and the new hybrid system (OG + Joint). Using the new NN architecture along with the new features yielded a 1% increase in the overall F1-score compared to the former hybrid system. Looking at specific entity types, the new hybrid system outperforms the other two systems in four out of the seven entity types. The new hybrid system achieves better F1-scores due to more balanced precision (65%) and recall scores (79%), while the former hybrid system has high precision (88%), but a lower recall (58%).

CR evaluation

In the evaluation at the level of Concept Recognition, a prediction was seen as correct if a ground-truth annotation existed at the same position with the same concept identifier. Again, we required the spans to be identical. Table 3 shows the performance of the knowledge-based system (OG), the previous hybrid system (OG + Dist) and the new hybrid system (OG + Joint) with respect to micro-averaged precision, recall and F1-scores in a strict evaluation scheme (no credit for partially overlapping spans). The overall F1-score of the new hybrid system (OG + Joint) improved by 7% compared to the previous hybrid system (OG + Dist). The difference is even more pronounced for the knowledge-based system (+ 27%). The higher F1-score increased mostly due to a much better overall precision (+ 14%), while the overall recall score only improved by 1%. In total, the new hybrid system outperforms the previous one in three and ties with four out of the seven entity types in terms of F1-scores.

Error analysis

Most false positives (FPs) are introduced by the aggressive matching algorithm of OGER. For example, the match 'IOP) [1' is returned for the string 'elevated intraocular pressure (IOP) [1–5]', as its collapsed form 'IOP1' is present in the terminologies. Another example is 'at 1', which is extracted from the string 'at 1 minute'

	Chemical	Sequence	Gene/protein	Cellular component	Cell	Cell line	Organ/tissue	Molecular function	Biological process	Organism	Disease
Chemical		7.35%	18.69%	4.46%	0.26%	2.79%	2.75%	3.74%	0.15%	0.90%	1.76%
Sequence	23.70%		11.77%	15.38%	0.00%	1.69%	2.07%	4.18%	0.29%	1.09%	0.30%
Gene/protein	28.82%	5.63%		4.27%	0.84%	8.97%	4.40%	7.98%	0.41%	4.98%	5.24%
Cellular component	29.72%	31.80%	18.43%		16.74%	1.56%	8.08%	4.39%	0.60%	0.22%	1.28%
Cell	3.98%	0.00%	8.33%	38.62%		4.33%	2.99%	0.82%	0.45%	0.02%	3.63%
Cell line	18.67%	3.52%	38.98%	1.57%	1.89%		5.73%	2.08%	0.18%	3.19%	6.30%
Organ/tissue	5.27%	1.23%	5.47%	2.33%	0.37%	1.64%		1.00%	1.16%	0.61%	1.28%
Molecular function	35.00%	12.16%	48.45%	6.18%	0.50%	2.91%	4.89%		0.65%	1.08%	2.21%
Biological process	0.83%	0.48%	1.46%	0.49%	0.16%	0.15%	3.30%	0.38%		0.03%	3.77%
Organism	3.71%	1.40%	13.34%	0.13%	0.01%	1.97%	1.32%	0.48%	0.02%		0.38%
Disease	4.46%	0.24%	8.64%	0.49%	0.60%	2.39%	1.69%	0.60%	1.75%	0.24%	

Fig. 6 Name overlap among different entity types. The figures in each row denote the percentage of names with this type that are also annotated with the type of the respective column. For example, of all mentions annotated as cell line, close to 39% also have a gene/protein annotation, while only 9% of the gene-annotated mentions also have an annotation as cell line

Table 2 Evaluation at the level of NER

Entity type	Precision			Recall			F1		
	OG	OG + Dist	OG + Joint	OG	OG + Dist	OG + Joint	OG	OG + Dist	OG + Joint
All	0.44	0.88	0.800	0.62	0.58	0.645	0.51	0.70	0.714
Chemicals	0.44	0.89	0.870	0.73	0.68	0.726	0.55	0.77	0.792
Cells	0.88	0.88	0.738	0.77	0.67	0.748	0.80	0.76	0.743
BPMFs	0.39	0.78	0.628	0.25	0.22	0.349	0.30	0.35	0.449
Cellular components	0.51	0.91	0.867	0.60	0.56	0.658	0.55	0.70	0.748
Organisms	0.29	0.98	0.977	0.92	0.91	0.920	0.44	0.94	0.948
Proteins	0.49	0.86	0.778	0.84	0.75	0.812	0.62	0.80	0.795
Sequences	0.46	0.89	0.833	0.67	0.64	0.670	0.54	0.75	0.743

because the term 'AT-1' has the normalized form 'at 1'. The postfilter fails to remove these two cases because the NN is largely trained on single words as input and only receives multi-word terms if it denotes a ground-truth entity. Thus, it never observes multi-word examples that are labeled as non-biomedical and learns that multi-word terms are always relevant. Another source of error are terms that are located within a word. For instance, the word 'Thr164Ala' contains the terms 'Thr' and 'Ala-' (normalized as 'Ala'). Some FPs are also common words such as 'processes' and 'positions' that also occur in terminologies and a small number are wrong re-classifications of the same span by the postfilter.

Most false negatives (FNs) are also caused by the knowledge-based system. While the postfilter can remove all types of FPs, it can only rectify FNs with the same span through re-classification, but not FNs with diverging spans, as these are pre-determined by the knowledge-based system. The vast majority of FNs are terms that are not listed verbatim in the terminologies:

- Morphological variations of the terms, e.g. 'carbonic' (→ 'carbon'), 'mammalian' (→ 'Mammalia')
- Abbreviations, e.g. 'bp' (→ 'base pair'), 'Chr' (→ 'chromosome')
- Synonyms, e.g. 'blood flow' (→ 'blood circulation'), 'chow' (→ 'food')
- Ellipses, e.g. 'A to G' (→ 'A to G transition'), 'alteration' (→ 'sequence alteration')
- Hyponyms, e.g. 'depression' (→ 'negative regulation of biological process'), 'passes' (→ 'establishment of localization').

Terms linked via the hyponym-hyperonym relation make up the largest group of these FNs and are pervasive for biological processes and molecular functions, whose recall is accordingly very low.

Conclusions

We have presented a fast, efficient, reliable entity NER-CR system for biomedical scientific literature. Competitive performance has been demonstrated by participation in a shared task and separate evaluations presented in this paper.

Besides fixing some of the remaining problems revealed by the error analysis presented in this paper, we are also

Table 3 Evaluation at the level of concept recognition

Entity type	Precision			Recall			F1		
	OG	OG + Dist	OG + Joint	OG	OG + Dist	OG + Joint	OG	OG + Dist	OG + Joint
All	0.32	0.51	0.650	0.52	0.49	0.503	0.40	0.50	0.567
Chemicals	0.28	0.59	0.601	0.61	0.57	0.568	0.39	0.58	0.584
Cells	0.88	0.87	0.878	0.72	0.66	0.713	0.79	0.75	0.787
BPMFs	0.35	0.72	0.634	0.19	0.17	0.178	0.25	0.27	0.278
Cellular components	0.49	0.87	0.930	0.59	0.56	0.581	0.54	0.68	0.716
Organisms	0.16	0.49	0.486	0.71	0.70	0.709	0.26	0.58	0.577
Proteins	0.45	0.84	0.788	0.83	0.74	0.799	0.59	0.79	0.794
Sequences	0.27	0.59	0.561	0.53	0.51	0.516	0.36	0.54	0.537

currently extending our experiments to multiple corpora, with different annotation strategies, with the goal of achieving competitive performance on several of them using a common architecture. We are also experimenting with more complex neural networks for the filtering stage, in particular recurrent NNs.

Abbreviations

API: application programming interface; BPMF: biological processes and molecular functions; ChEBI: chemical entities of biological interest; CR: concept recognition; CRAFT: Colorado Richly Annotated Full Text; CRF: conditional random fields; CTD: Comparative Toxicogenomics Database; FN: false negative; FP: false positive; HTTP: Hypertext Transfer Protocol; JSON: JavaScript Object Notation; MeSH: Medical Subject Headings; NCBI: US National Center for Biotechnology Information; NER: named entity recognition; NLP: natural language processing; NLTK: Natural Language Toolkit; NN: neural network; OGER: OntoGenes entity recognizer; PMC: PubMed Central; REST: Representational State Transfer; TIPS: Technical interoperability and performance of annotation servers; URL: Unified Resource Locator; XML: Extensible Markup Language.

Authors' contributions

LF implemented the dictionary-based annotator and the integration of all components in the web service, carried out the speed benchmarks and provided support for the NER/CR experiments. AJ re-implemented the NN classifier and performed the CRAFT-based evaluation. NC assisted in the creation of various components. FR provided advice, guidance, and support. All authors read and approved the final manuscript.

Author details

[1] Institute of Computational Linguistics, University of Zurich, Andreasstr. 15, 8050 Zürich, Switzerland. [2] Fondazione Bruno Kessler, Via Sommarive, 18, 38123 Trento, Italy.

Acknowledgements

The authors wish to express their gratitude to the organizers of the TIPS challenge, which offered an opportunity to benchmark OGER against a number of competing systems. Additional thanks to the reviewers for their helpful suggestions.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The CRAFT corpus (version 2.0) used in the current study is available from http://bionlp-corpora.sourceforge.net/CRAFT/. The source code of OGER is available from https://github.com/OntoGene/OGER. The source code of the NN postfilter (technically a plug-in for OGER) is available from https://github.com/OntoGene/OGER-filter.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Funding

The research activities of the OntoGene/BioMeXT group at the University of Zurich are supported by the Swiss National Science Foundation (Grant CR30I1_162758).

Publisher's Note

Received: 31 July 2018 Accepted: 27 December 2018
Published online: 21 January 2019

References

1. Fukuda K-I, Tsunoda T, Tamura A, Takagi T (1998) Toward information extraction: identifying protein names from biological papers. In: Pacific symposium on biocomputing, vol 3, pp 705–716
2. Kemp N, Lynch M (1998) Extraction of information from the text of chemical patents. 1. Identification of specific chemical names. J Chem Inf Comput Sci 38(4):544–551. https://doi.org/10.1021/ci980324v
3. Narayanaswamy M, Ravikumar KE, Vijay-Shanker K (2003) A biological named entity recognizer. In: Pacific symposium on biocomputing, vol 8, pp 427–438
4. Leaman R, Gonzalez G (2008) BANNER: an executable survey of advances in biomedical named entity recognition. In: Pacific symposium on biocomputing, vol 13, pp 652–663
5. Klinger R, Kolářik C, Fluck J, Hofmann-Apitius M, Friedrich CM (2008) Detection of IUPAC and IUPAC-like chemical names. Bioinformatics 24(13):268–276. https://doi.org/10.1093/bioinformatics/btn181
6. Campos D, Matos S, Oliveira JL (2013) Gimli: open source and high-performance biomedical name recognition. BMC Bioinform 14:54. https://doi.org/10.1186/1471-2105-14-54
7. Kaewphan S, Van Landeghem S, Ohta T, Van de Peer Y, Ginter F, Pyysalo S (2016) Cell line name recognition in support of the identification of synthetic lethality in cancer from text. Bioinformatics 32(2):276–282. https://doi.org/10.1093/bioinformatics/btv570
8. Tanenblatt M, Coden A, Sominsky I (2010) The ConceptMapper approach to named entity recognition. In: Calzolari N, Choukri K, Maegaard B, Mariani J, Odijk J, Piperidis S, Rosner M, Tapias D (eds) Proceedings of the seventh international conference on language resources and evaluation (LREC'10). European Language Resources Association (ELRA)
9. Hakenberg J, Gerner M, Haeussler M, Solt I, Plake C, Schroeder M, Gonzalez G, Nenadic G, Bergman CM (2011) The GNAT library for local and remote gene mention normalization. Bioinformatics 27(19):2769–2771. https://doi.org/10.1093/bioinformatics/btr455
10. Bravo À, Cases M, Queralt-Rosinach N, Sanz F, Furlong LI (2014) A knowledge-driven approach to extract disease-related biomarkers from the literature. BioMed Res Int 2014:253128. https://doi.org/10.1155/2014/253128
11. Tseytlin E, Mitchell K, Legowski E, Corrigan J, Chavan G, Jacobson RS (2016) NOBLE–flexible concept recognition for large-scale biomedical natural language processing. BMC Bioinform 17(1):1–15. https://doi.org/10.1186/s12859-015-0871-y
12. Aronson AR, Lang F-M (2010) An overview of MetaMap: historical perspective and recent advances. J Am Med Inform Assoc 17(3):229–236. https://doi.org/10.1136/jamia.2009.002733
13. Savova GK, Masanz JJ, Ogren PV, Zheng J, Sohn S, Kipper-Schuler KC, Chute CG (2010) Mayo clinical text analysis and knowledge extraction system (cTAKES): architecture, component evaluation and applications. J Am Med Inform Assoc 17(5):507–513. https://doi.org/10.1136/jamia.2009.001560
14. Leaman R, Islamaj Doğan R, Lu Z (2013) DNorm: disease name normalization with pairwise learning to rank. Bioinformatics 29(22):2909–2917. https://doi.org/10.1093/bioinformatics/btt474
15. Pathak P, Patel P, Panchal V, Soni S, Dani K, Patel A, Choudhary N (2015) ezDI: a supervised NLP system for clinical narrative analysis. In: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), pp 412–416. Association for Computational Linguistics
16. Cuzzola J, Jovanović J, Bagheri E (2017) RysannMD: a biomedical semantic annotator balancing speed and accuracy. J Biomed Inform 71:91–109. https://doi.org/10.1016/j.jbi.2017.05.016
17. Sasaki Y, Tsuruoka Y, McNaught J, Ananiadou S (2008) How to make the most of NE dictionaries in statistical NER. BMC Bioinform 9(11):5. https://doi.org/10.1186/1471-2105-9-S11-S5
18. Rocktäschel T, Weidlich M, Leser U (2012) ChemSpot: a hybrid system for chemical named entity recognition. Bioinformatics 28(12):1633–1640. https://doi.org/10.1093/bioinformatics/bts183

19. Leaman R, Wei C-H, Lu Z (2015) tmChem: a high performance approach for chemical named entity recognition and normalization. J Cheminform 7(Suppl 1):3. https://doi.org/10.1186/1758-2946-7-S1-S3
20. Akhondi SA, Pons E, Afzal Z, van Haagen H, Becker BFH, Hettne KM, van Mulligen EM, Kors JA (2016) Chemical entity recognition in patents by combining dictionary-based and statistical approaches. Database 2016. https://doi.org/10.1093/database/baw061
21. Munkhdalai T, Li M, Batsuren K, Park HA, Choi NH, Ryu KH (2015) Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations. J Cheminform 7(1):9. https://doi.org/10.1186/1758-2946-7-S1-S9
22. Lee HC, Hsu YY, Kao HY (2015) An enhanced CRF-based system for disease name entity recognition and normalization on BioCreative V DNER task. In: Proceedings of the fifth biocreative challenge evaluation workshop, pp 226–233
23. Leaman R, Lu Z (2016) TaggerOne: joint named entity recognition and normalization with semi-Markov models. Bioinformatics 32(18):2839. https://doi.org/10.1093/bioinformatics/btw343
24. ter Horst H, Hartung M, Cimiano P (2017). In: Gracia J, Bond F, McCrae JP, Buitelaar P, Chiarcos C, Hellmann S (eds) Joint entity recognition and linking in technical domains using undirected probabilistic graphical models, vol 10318, pp 166–180. Springer, Cham. https://doi.org/10.1007/978-3-319-59888-8_15
25. Jiang Z, Li L, Huang D, Jin L (2015) Training word embeddings for deep learning in biomedical text mining tasks. In: 2015 IEEE international conference on bioinformatics and biomedicine (BIBM), pp 625–628. https://doi.org/10.1109/BIBM.2015.7359756
26. Li F, Zhang Y, Zhang M, Ji D (2016) Joint models for extracting adverse drug events from biomedical text. In: Proceedings of the twenty-fifth international joint conference on artificial intelligence (IJCAI-16), pp 2838–2844
27. Li F, Zhang M, Fu G, Ji D (2017) A neural joint model for entity and relation extraction from biomedical text. BMC Bioinform 18(1):198. https://doi.org/10.1186/s12859-017-1609-9
28. Crichton G, Pyysalo S, Chiu B, Korhonen A (2017) A neural network multi-task learning approach to biomedical named entity recognition. BMC Bioinform 18(1):368. https://doi.org/10.1186/s12859-017-1776-8
29. Luo L, Yang Z, Yang P, Zhang Y, Wang L, Lin H, Wang J (2018) An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. Bioinformatics 34(8):1381–1388. https://doi.org/10.1093/bioinformatics/btx761
30. Li H, Chen Q, Tang B, Wang X, Xu H, Wang B, Huang D (2017) CNN-based ranking for biomedical entity normalization. BMC Bioinform 18(11):385. https://doi.org/10.1186/s12859-017-1805-7
31. Furrer L, Rinaldi F (2017) OGER: OntoGene's entity recogniser in the BeCalm TIPS task. In: Proceedings of the BioCreative V.5 challenge evaluation workshop, pp 175–182
32. Basaldella M, Furrer L, Tasso C, Rinaldi F (2017) Entity recognition in the biomedical domain using a hybrid approach. J Biomed Semant 8(1):51
33. OpenMinTeD. http://openminted.eu/. Accessed 25 July 2018
34. Comeau DC, Islamaj Doğan R, Ciccarese P, Cohen KB, Krallinger M, Leitner F, Lu Z, Peng Y, Rinaldi F, Torii M et al (2013) BioC: a minimalist approach to interoperability for biomedical text processing. Database 2013 . https://doi.org/10.1093/database/bat064
35. Kim JD, Wang Y (2012) PubAnnotation: a persistent and sharable corpus and annotation repository. In: Proceedings of the 2012 workshop on biomedical natural language processing, pp 202–205. Association for Computational Linguistics
36. PubAnnotation: Annotation format. http://www.pubannotation.org/docs/annotation-format/. Accessed 25 July 2018
37. Rinaldi F, Kappeler T, Kaljurand K, Schneider G, Klenner M, Clematide S, Hess M, von Allmen J-M, Parisot P, Romacker M, Vachon T (2008) OntoGene in BioCreative II. Genome Biol 9(2):13
38. Rinaldi F, Schneider G, Kaljurand K, Clematide S, Vachon T, Romacker M (2010) OntoGene in BioCreative II.5. IEEE/ACM Trans Comput Biol Bioinform 7(3):472–480
39. Rinaldi F, Clematide S, Hafner S (2012) Ranking of CTD articles and interactions using the OntoGene pipeline. In: Proceedings of the 2012 BioCreative workshop, Washington, DC
40. Rinaldi F, Clematide S, Marques H, Ellendorff T, Rodriguez-Esteban R, Romacker M (2014) OntoGene web services for biomedical text mining. BMC Bioinform 15(14):S6
41. Basaldella M, Furrer L, Colic N, Ellendorff TR, Tasso C, Rinaldi F (2016) Using a hybrid approach for entity recognition in the biomedical domain. In: Neves M, Rinaldi F, Nenadic G, Rebholz-Schuhmann D (eds) Proceedings of the 7th international symposium on semantic mining in biomedicine, pp 11–19
42. Bada M, Eckert M, Evans D, Garcia K, Shipley K, Sitnikov D, Baumgartner WA, Cohen KB, Verspoor K, Blake JA (2012) Concept annotation in the CRAFT corpus. BMC Bioinform 13(1):161
43. Degtyarenko K, De Matos P, Ennis M, Hastings J, Zbinden M, McNaught A, Alcántara R, Darsow M, Guedj M, Ashburner M (2008) ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Res 36(suppl 1):344–350
44. Cell Ontology: an ontology of cell types. http://obofoundry.org/ontology/cl.html. Accessed 10 July 2018
45. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT (2000) Gene Ontology: tool for the unification of biology. Nat Genet 25(1):25–29
46. NCBI Taxonomy. http://www.ncbi.nlm.nih.gov/taxonomy. Accessed 10 July 2018
47. Protein Ontology. http://pir.georgetown.edu/pro/pro.shtml. Accessed 10 July 2018
48. Eilbeck K, Lewis SE, Mungall CJ, Yandell M, Stein L, Durbin R, Ashburner M (2005) The Sequence Ontology: a tool for the unification of genome annotations. Genome Biol 6(5):44
49. Bairoch A (2018) The Cellosaurus, a cell-line knowledge resource. J Biomol Tech 29(2):25–38. https://doi.org/10.7171/jbt.18-2902-002
50. Davis AP, Murphy CG, Saraceni-Richards CA, Rosenstein MC, Wiegers TC, Mattingly CJ (2009) Comparative toxicogenomics database: a knowledgebase and discovery tool for chemical–gene–disease networks. Nucleic Acids Res 37(suppl 1):786–792. https://doi.org/10.1093/nar/gkn580
51. Lipscomb CE (2000) Medical Subject Headings (MeSH). Bull Med Libr Assoc 88(3):265–266
52. The UniProt Consortium: the universal protein resource (UniProt). Nucleic Acids Res 36(suppl 1):190–195 (2008). https://doi.org/10.1093/nar/gkm895
53. Mungall CJ, Torniai C, Gkoutos GV, Lewis SE, Haendel MA (2012) Uberon, an integrative multi-species anatomy ontology. Genome Biol 13(1):5. https://doi.org/10.1186/gb-2012-13-1-r5
54. Ellendorff TR, Van der Lek A, Furrer L, Rinaldi F (2015) A combined resource of biomedical terminology and its statistics. In: Proceedings of the 11th international conference on terminology and artificial intelligence, pp 39–50
55. Jancso A (2018) Using a neural network to correct the output of a lexicon-based NER system. Bachelor's thesis, University of Zurich, Switzerland
56. Hunspell. http://hunspell.github.io/. Accessed 25 July 2018
57. Bird S, Loper E (2004) NLTK: the natural language toolkit. In: Proceedings of the ACL interactive poster and demonstration sessions
58. Chiu B, Crichton GKO, Korhonen A, Pyysalo S (2016) How to train good word embeddings for biomedical NLP. In: Proceedings of the 15th workshop on biomedical natural language processing, pp 166–174
59. Sayers E (2009) The E-utilities in-depth: parameters, syntax and more. Entrez Programming Utilities Help. Bethesda (MD): National Center for Biotechnology Information (US); 2010. https://www.ncbi.nlm.nih.gov/books/NBK25499/. Updated 1 Nov 2017
60. Pérez-Pérez M, Pérez-Rodríguez G, Blanco-Míguez A, Fdez-Riverola F, Valencia A, Krallinger M, Lourenco A (2017) Benchmarking biomedical text mining web servers at BioCreative V.5: the technical interoperability and performance of annotation servers—TIPS track. In: Proceedings of the BioCreative V.5 challenge evaluation workshop, pp 12–21