Learning Representations for Biomedical Named Entity Recognition

Ivano Lauriola[1,2], Riccardo Sella[1], Fabio Aiolli[1], Alberto Lavelli[2], and Fabio Rinaldi[2,3]

[1] University of Padova - Department of Mathematics
Via Trieste, 63, 35121 Padova - Italy

[2] Fondazione Bruno Kessler
Via Sommarive, 18, 38123 Trento - Italy

[3] University of Zurich - Institute of Computational Linguistics
Andreasstrasse 15, CH-8050 Zurich - Switzerland

ivano.lauriola@phd.unipd.it

Abstract. Biomedical Named Entity Recognition is a common task in Natural Language Processing applications, whose purpose is to recognize and categorize different types of entities in biomedical documents. Recently, the literature has shown effective methods based on combinations of Machine Learning algorithms and Natural Language Processing techniques. However, a critical issue of such applications is the choice of the data representation. Generic and abstract word-embeddings can be easily used to train a learning algorithm, without prior knowledge of the domain. On the other hand, dedicated hand-crafted features are expensive to define, but they could represent better the specific problem.
In this work, an extensive experimental assessment is carried out, where different representations have been analyzed. Then, a general framework to learn the representation by combining general and domain-specific features is proposed and evaluated, showing empirical results on the CRAFT corpus.

Keywords: Named Entity Recognition, Representation learning, Multiple Kernel Learning

1 Introduction

The constant growth of the biomedical literature requires increasingly complex methods to index, categorize and retrieve documents from large-scale online repositories. The aim of Biomedical Named Entity Recognition (BNER) is to recognize and extract relevant entities and concepts from the biomedical literature. These entities can be the name of proteins, cellular components, diseases, species and so on, and they could help large-scale searching algorithms to retrieve relevant documents.
One of the main difficulties of this task is the ambiguity of the terms. A single

term can refer to different concepts. A classical example is provided by the token *CAT*, which can refer to an animal, or it can be the acronym for *Computed Aided Tomography* or for *Chloramphenicol Acetyl Transferase*. Another source of difficulties is that proteins and other biomedical entities can be written in different ways (e.g. HIV-1 versus HIV 1).

Natural Language Processing (NLP) techniques have been widely used in the literature to solve these tasks [20]. Standard approaches include the usage of human-designed rules applied on the document, or exact match with a dictionary which contains all possible entities. However, there are some issues with these methods, such as the human effort to handle and update the dictionary, and the difficulty of designing powerful and expressive rules.

Recently, Machine Learning algorithms have been combined with standard NLP techniques [8], aiming to improve the performance of these systems. State of the art methods include the application of Deep Neural Networks, focusing on 1D Convolutional Neural Network (CNN) and Long-Short Term Memory (LSTM).

One of the main issues on the application of machine learning algorithms on the BNER task is the choice of the data representation which describes tokens and entities. It is shown in the literature [6] that different representations emphasize different aspects of the problem, and they provide different results. Hence, the selection of the representation is a key aspect for a powerful predictor. Several representations have been analyzed in the literature to solve BNER tasks, each of them defining a particular point of view of the main problem.

In [5], a set of hand-crafted and domain-specific character-level features have been considered. These features describe the inner structure of tokens, such as the number and position of upper and lower characters, the affixes, the presence of symbols and so on. The idea behind this representation is that biomedical entities have a particular inner structure easily recognizable by the defined characteristics. In other works, more general representations based on word embeddings have been used to represent the tokens (see [24, 11]), reducing the human effort on the feature engineering phase, and making easier to adapt these systems to new biomedical entity types. However, these representations are not able to solve the disambiguation problem, since they consider only the character-level features. Hence, the same words have the same representation independently of their position in the text.

On the other hand, word-level representations consider the spatial and semantic information of tokens and entities in the document, aiming to solve the disambiguation problem. These representations consider the position of the entity with respect to the other tokens, or the other entities.

The main contribution of this work is an extensive analysis and comparison of different data representations in the BNER task, where each of them emphasizes different viewpoints of the problem, and corresponds to different abstraction levels. Then, a general framework based on the Multiple Kernel Learning paradigm is proposed to learn the best representation from the training data directly.

Several baselines based on deep and shallow machine learning techniques have been compared with the proposed method, showing its empirical effectiveness in terms of efficacy, measured by means of the F_1 score.

The paper is organized as follows. Section 2 provides a background on the BNER task, including a description of the Multiple Kernel Learning paradigm and the related work. Then, the proposed method is defined in Section 3. Eventually, Section 4 contains the experimental assessment and the results reached by the baselines and the proposed method.

2 Background and Related work

NLP applications rely on a sequence of steps that extract structured textual features from the document. Usually, the first step is to divide the text into sentences (sentence splitting) and the sentences into tokens (tokenization). Additional normalization steps can follow at the token level, such as determining the lexical root of words (lemmatization). Through morpho-syntactic analysis it is then possible to determine the part of speech of words (e.g. noun, verb, adjective).

NER can be performed either on general texts (e.g., newspaper articles), to recognize concepts like person, organization or location, or on technical documents (e.g., biomedical literature), to recognize concepts like cells, diseases or proteins. NER can be used by itself, with the goal of recognizing the presence of a term in a given document, or as a preliminary step for further, more complex tasks (e.g., relation extraction).

Several approaches exist in the literature to solve the NER task. They can be grouped in the following categories:

- Rule-based: these methods consist of domain-specific hand-written rules which are able to recognize entities in documents. The rules consider regular expressions or particular characteristics of the entities. Generally, these rules are defined by groups of biomedical and linguistic experts.
- Dictionary-based: the simplest approach, which finds the occurrences of entities in a document from a precompiled dictionary or ontology, which contains all of the possible entities. However, the maintenance and the constant update of dictionaries from specific domains is an expensive task.
- Machine Learning methods: shallow machine learning techniques have been widely applied on the NER/BNER task, such as the Support Vector Machine (SVM), Conditional Random Field (CRF) and Hidden Markov Models (HMM), showing good results with domain-specific features. Recently, deep learning algorithms have been considered, like the 1D CNN and LSTM, with promising results.

The interest for the NER task in the biomedical domain has produced an extensive literature. Here we briefly discuss about the major advances.
In [25] *Conditional Random Fields* are used, together with handcrafted features, in order to improve previous state of the art results.
High-performance BNER systems often consider hybrid approaches, where rule-

or dictionary- based approaches and machine learning techniques are combined. A multiclass BNER problem has been analyzed in [18], where authors proposed a two-step algorithm, where in the first phase entities are recognized by means of the SVM algorithm. Then a dictionary look-up is applied to classify entities. Authors in [5] proposed a different hybrid approach, which consists of a dictionary look-up as first step and machine learning output filtering in the second step. This ensemble system is shown to empirically achieve state of the art results on the CRAFT corpus.

In [9] is described an extensive quantitative analysis about word vectors trained on millions of documents of the biomedical literature, suggesting that using word vectors could improve the results in various related tasks.

The claim of [9] found applications through the usage of a fairly complex model based on Long Short Term Memory deep neural networks and Conditional Random Fields [17, 10].

2.1 Multiple Kernel Learning

Kernel Machines are a large family of Machine Learning algorithms widely used in the literature to solve classification, regression and clustering problems.

A kernelized algorithm comprises two elements. The first element is the learning algorithm whose solution is expressed by dot-products between training examples. The second consists of a symmetric positive semi-definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which computes the dot-product in a Reproducing Kernel Hilbert Space (RKHS). This means that there is a function $\phi : \mathcal{X} \to \mathcal{K}$ which maps data from the input space $\mathcal{X}$ to the kernel space $\mathcal{K}$ such that $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$, where $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$. Usually, an expert user chooses the kernel function exploiting her/his domain-specific knowledge, or via a validation procedure.

Recently the literature showed mechanisms to learn the kernel function directly from the training data. The most well known kernel learning paradigm is the Multiple Kernel Learning (MKL) [16], which learns the kernel as a linear non-negative combination of P base kernels, with the form:

$$k_{\boldsymbol{\mu}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{r=1}^{P} \mu_r k_r(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad \mu_r \geq 0$$

where k_r is the r-th kernel function defined on the r-th representation ϕ_r, and $\boldsymbol{\mu}$ is the weights vector which parametrizes the combination. These P base kernels correspond to different source, or different notions of similarity between examples.

3 Method

This work considers/describes an extended version of the learning pipeline proposed in [5], which is a two-stage hybrid procedure to recognize entities in the biomedical literature. The next subsections describe this hybrid system (3.1) and the proposed one (3.2), emphasizing the differences and strengths.

3.1 A two-stage hybrid pipeline

Authors in [5] proposed a hybrid system which combines NLP and machine learning techniques to recognize entities from documents. The system acts by means of a two-stage pipeline.

The first phase of this pipeline consists of a dictionary-based filter, where a set of candidate entities is recognized from the corpus by means of dictionary look-up. The aim of the filter is to discard the large part of non-entities from documents, resulting in high recall but low precision.

Then, a feature vector is computed for each candidate by means of a hand-crafted representation, which considers a set of character level features and affixes. Eventually, a classifier based on neural networks is used to recognize entities from the set of candidates. Dictionary look-up is applied on both the training and test documents. Training candidates are used to train a machine learning algorithm.

There are two weaknesses in this approach. Firstly, the training set used to train the neural network is composed exclusively of the output of the dictionary-based classifier, which corresponds to the set of candidate entities. The positive class is composed by the candidates that correspond to annotated entities, whereas the remaining candidates form the negative class.

Besides, entities discarded by the dictionary filter are not used in the training phase, with a consequent loss of useful information. Moreover, when the first layer of the system works well, there is a further lack of negative examples. Hence, the application of complex Neural Networks is expected to result in lower performance.

3.2 The proposed extension

In order to overcome the above mentioned limitations, an extended training set is taken into consideration to train the machine learning algorithms. On the one hand, the whole set of annotated entities defines the positive examples. On the other hand, the negative set is composed by the False Positive candidates from the dictionary-based filter. Furthermore, additional negative examples/tokens have been included in the training set to reduce the lack of negative examples. These tokens consist of words that are not entities nor stop-words from the training corpus, that are discarded by the dictionary classifier, and they correspond to 50% of the positive examples.

Hence, if the corpus contains N annotated entities, and the dictionary filter provides a candidate set composed by $TP(< N)$ True Positive and FP False Positive entities, the dataset will contains N positive examples and $FP + \frac{N}{2}$ negative examples.

The main extension proposed in this work concerns the generalization of the pipeline, by including a mechanism to learn the best representation directly from data, exploiting the MKL framework.

Three different explicit representations have been considered as a descriptor of each candidate entity. The first representation consists of a word embedding computed by means of the Word2Vec algorithm [19]. The embedding has been

trained on the PubMed corpus, and it is available in the Gensim package for the python programming language [22].
Moreover, the hand-crafted representation defined in [5] has been used. This representation consists of two main groups of features. The former group contains features which focus on the affixes. The latter group describes the structure of the token from an orthographic point of view, i.e. the number of upper and lower characters, the presence of symbols and numbers... See [5] for the complete list of these features. In our work, these groups of features have been divided into two different representations to improve the expressiveness of the MKL algorithm. 5 Homogeneous Polynomial Kernels with degrees $\{1 \dots 5\}$ are computed for each representation. Eventually, a MKL algorithm provides the combination of these 15 kernels. The choice of such kernels derives from theoretical results on the generalization of dot-product kernels. See [13] to get more details.

Several MKL algorithms exists in the literature. In this work the EasyMKL [2] algorithm has been considered. EasyMKL is an efficient state-of-the-art MKL algorithm which tries to find the combination of base kernels that maximize the distance between the convex hull of the positive examples and the convex hull of the negative ones.

The proposed approach has two main advantages with respect to standard machine learning approaches. First of all, the scalability with respect to the number of representations. Generally, the computational complexity of MKL algorithms increases linearly with the number of kernels. This means that the adding of novel representation requires only the computation of the associated feature vector, which generally is not an expensive task.
Furthermore, the proposed procedure does not require the validation and the selection of the representation. The larger is the pool of representations, the more expressive is the MKL algorithm. A depiction of the proposed system is described in Fig. 1.

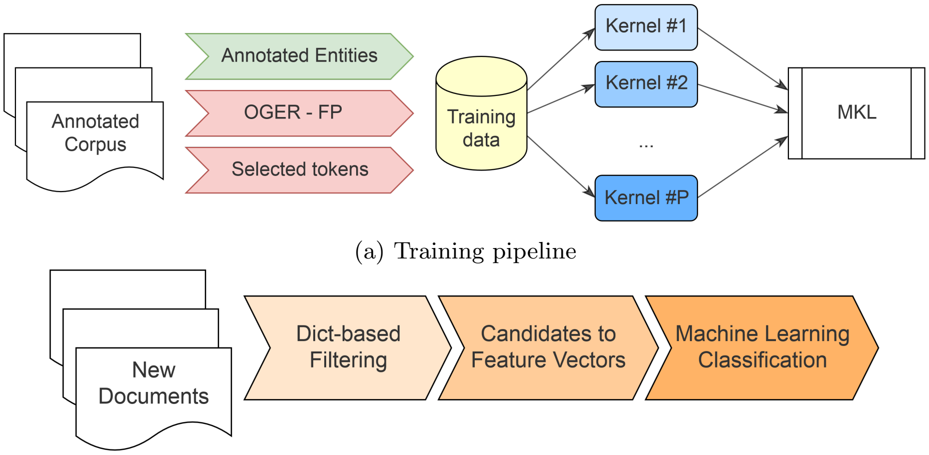
4 Experimental assessment

The whole set of experiments is described in this section. The dataset, the algorithms and some baselines are also discussed.

4.1 Dataset

The experimental analysis has been conducted on the Colorado Richly Annotated Full Text (CRAFT) corpus v2.0 [3]. The CRAFT corpus contains a set of 67 documents from the PubMed Central Open Access Subset. These documents have been manually annotated with respect to the following ontologies:

- Chemical Entities of Biological Interest (ChEBI) [12]: contains chemical names;
- Cell Ontology (CL) [4]: contains names of cell types;



(a) Training pipeline

(b) Test pipeline

Fig. 1: A depiction of the proposed system. During the traning phase (a) the MKL algorithm learn the weights of the linear non-negative combination of base kernels. Then (b), the learned representations are used to classify candidates by means of a two-stage pipeline.

- Gene Ontology (GO) [7]: the CRAFT corpus is annotated with two subcategory, which are Cellular Components (GO_CC) and Biological Processes and Molecular Functions (GO_BPMF);
- National Center for Biotechnology Information (NCBI) Taxonomy [15]: includes names of species and taxonomic ranks;
- Protein Ontology (PR) [1]: contains protein names;
- Sequence Ontology (SO) [14]: contains names of biological sequence features and attributes.

The corpus includes 570 000 tokens, with approximately 100 000 annotated concepts and more than 21 000 sentences. The 7 ontologies have been analyzed individually by using both the base training set used in [5], and the extended one, discussed in the previous section.

4.2 Baselines

Several hard baselines have been analyzed and compared:

- Multiple Layer Perceptron (MLP): the approach proposed in [5], with the same architecture.
- Random Forest (RF): due to its generalization capability in several domains, and its computational efficiency, the RF classifier has been considered as a further baseline.

– Convolutional Neural Network (CNN): this algorithm has been considered aiming to combine the character-level features to the context and semantic information.

MLP and RF algorithms consider exclusively the information that the candidate entity provides, that is its representation. CNN instead, considers a small-sized window of tokens around the candidate. Hence it has more information with respect to the other baselines, which can be used to improve the overall classification performance, reducing the disambiguation problem.

4.3 Evaluation

The CRAFT corpus has been divided in training (47) and test (20) documents, by considering the same split used in [5].
The dictionary-based classifier has been used on the training corpus to produce the training sets and on the test corpus to recognize the candidate set. Both the base training set ([5]) and the extended training set (see Section 3) have been considered individually.
To accomplish this step, the Onto-Gene's Entity Recognizer (OGER) [23] framework has been used. Each document has been split in tokens, by using the lossy tokenization method of splitting every time a non-alphanumeric character is found. Each token has been then converted to lowercase, and stemming (using Lancaster stemmer) has been applied, except for acronyms. Greek letters have been expanded ($\alpha \rightarrow alpha$), with the aim of further normalizing the final tokens.

The details and results of the first phase of the architecture, which corresponds to dictionary look-up, are shown in the Table 1, including the number of entities for each ontology, the number of candidates, of True Positives and of False Positives on both the training and the test corpus.

A hold-out procedure has been applied by splitting the training set in training (80%) and validation/development (20%) to select the hyperparameters, which are:

MLP : the architecture presented in [5] has been used. No additional hyperparameters have been validated. The validation set has been used to prevent the overfitting by means of an early-stop procedure.

RF : the number of trees used, with values $\{10, 50, 100, 200, 500, 1000\}$. Other hyperparameters have been set to their default values, defined in the Scikit-learn implementation [21].

1D-CNN : the number of convolutional layers, from 1 up to 4, each of them with 128 filters. The dimension of the window around the candidate has been also validated, starting from 5 tokens up to 14.

EasyMKL : the λ value of the algorithm, with values $\{0.0, 0.1, \ldots, 1.0\}$. This value regularizes the solution by maximizing the distance between centroids which represent the classes rather than the margin. A hard-margin SVM has been used as base learner.

Table 1: Detailed results of the dictionary filter, including the number of candidate entities, true positives and false positives computed on the training documents (first row) and test documents (second row). F_1 (precision,recall) scores are also reported.

	# annotated entities	# candidates	TP	FP	pos/neg (%)	F_1
ChEBI	5736	9284	4033	5251	43/57	54 (43,70)
	1800	3020	1319	1710		55 (44,73)
CL	4612	3804	3423	381	90/10	81 (90,75)
	1266	1044	923	121		80 (88,73)
GO_BPMF	15608	10870	3821	7049	35/65	29 (35,25)
	5608	3573	1377	2196		30 (39,25)
GO_CC	6302	7457	4419	3038	59/41	64 (59,70)
	2075	2431	1236	1195		55 (51,60)
NCBI	5432	17696	4832	12864	27/73	42 (27,89)
	2021	6312	1854	4458		44 (30,92)
PR	11827	19240	9599	9641	50/50	62 (50,81)
	3814	6502	3199	3303		62 (49,84)
SO	15143	24027	11093	12934	46/54	57 (46,73)
	6093	8796	4056	4740		54 (46,67)
all	87337	124056	55184	68881	44/56	52 (44,63)

In order to find entities in the test documents, the dictionary-based classification has been performed to find the candidate set. Then, the trained machine learning algorithm has been applied to further classify the examples.
Algorithms and representations have been compared by considering precision, recall and F_1. Results reached by the baselines on the two representations and the proposed method are shown in Table 2. Results are also summarized in the Table 3, where the average rank of each baseline is shown.

4.4 Discussion

Several algorithms have been analyzed in this work. The MLP architecture proposed in [5] provides lower results with respect to the RF algorithm, whose training is less expensive by orders of magnitude. A notable result is the low F_1 reached by the deep CNN, which was the most favourite algorithm.

It is clear that each method has its own suitable representation, which is the hand-crafted for the RF algorithm, and the embedding computed by the Word2Vec algorithm for the Neural Networks. Besides, the MKL approach achieves high results avoiding the selection of the representation. However, in this work only 3 types of kernels have been taken into consideration, bounding the expressiveness of the proposed approach.

The inclusion of additional training examples provides an empirical improvement of the overall performance, with a general increment of 0-2% points of F_1 score.
Eventually, the experimental assessment on the MLP algorithm confirms the re-

Table 2: F_1 (precision,recall) scores computed on the NER classification task by using different representations. For each ontology, the first row considers ML models trained on the output of the dictionary, whereas the second row considers ML models trained on the extended training set.

	[5]			word2vec			
	MLP	RF	CNN	MLP	RF	CNN	MKL
ChEBI	76 (89,66)	79 (92,69)	80 (95,70)	78 (87,70)	78 (87,70)	75 (87,66)	79 (91,70)
	77 (88,69)	80 (92,70)	80 (94,70)	79 (89,71)	78 (87,70)	75 (87,66)	80 (95,70)
CL	76 (87,67)	81 (89,74)	83 (98,72)	81 (89,74)	81 (89,74)	83 (96,74)	80 (89,74)
	78 (90,69)	82 (90,75)	83 (96,72)	81 (89,75)	81 (89,74)	84 (96,74)	80 (89,73)
GO_BPMF	35 (67,24)	36 (80,23)	31 (57,22)	35 (71,24)	36 (72,24)	36 (79,23)	36 (79,23)
	36 (70,24)	37 (85,24)	38 (65,27)	37 (73,25)	36 (72,24)	36 (80,23)	36 (72,24)
GO_CC	70 (92,56)	68 (92,54)	44 (39,50)	69 (87,57)	67 (89,54)	70 (88,57)	69 (86,57)
	70 (92,56)	69 (93,54)	47 (44,50)	69 (87,57)	70 (87,57)	69 (86,57)	70 (88,57)
NCBI	94 (98,91)	95 (99,91)	94 (99,88)	91 (90,92)	90 (89,91)	95 (98,92)	95 (99,91)
	94 (98,91)	95 (99,91)	95 (99,91)	90 (90,89)	90 (89,91)	94 (98,91)	95 (99,91)
PR	80 (87,74)	83 (86,80)	88 (94,83)	77 (80,74)	79 (81,77)	80 (88,74)	82 (88,76)
	81 (88,75)	83 (89,78)	88 (95,83)	80 (83,76)	80 (82,77)	81 (89,74)	82 (88,77)
SO	75 (92,63)	75 (93,63)	72 (78,64)	74 (92,62)	75 (91,63)	75 (92,63)	75 (93,63)
	74 (92,62)	75 (93,63)	72 (79,64)	75 (92,64)	75 (91,63)	76 (93,65)	75 (92,63)

Table 3: Average rank of F_1, precision and recall scores reached by the algorithms by using the base (first row), and the extended training set (second row).

	[5]			word2vec			
	MLP	RF	CNN	MLP	RF	CNN	MKL
F_1	4.57	**2.71**	4.00	4.57	4.71	3.14	**2.71**
	4.86	**2.86**	3.00	4.28	4.85	4.28	3.00
precision	4.14	**2.00**	3.57	4.86	4.86	3.14	2.57
	3.71	**1.71**	3.57	4.43	5.58	3.14	3.00
recall	4.00	3.14	4.29	2.43	**2.29**	2.86	**2.29**
	5.00	2.71	2.86	2.71	**2.43**	3.86	2.71

sults reached in [5], with the exception of the ChEBI ontology, where we reach 2 points less than the previous work. This difference could depend on the random component of the optimizer used.

5 Conclusions

In this work, a general framework for learning the best representation for the Biomedical Named Entity Recognition task is presented and analyzed. The proposed method aims to combine several weak representations in a single one by means of the Multiple Kernel Learning paradigm. These representations define different points of view, and emphasize different aspects of the problem through a different set of features.

An empirical evaluation against hard baselines has been performed, showing the generalization capability of the proposed framework on the CRAFT corpus, with promising results.

In the future, we plan to extend the proposed approach in different directions. Firstly, more representations will be included, such as Word2Vec models pre-trained on different domains (e.g.: Wikipedia, GoogleNews...), character-level embeddings, word-normalization features, and Part-Of-Speech information. Secondly, the weights that the Multiple Kernel Learning algorithm assigns will be analyzed. We aim to understand which are the most relevant feature sets in the combination for each ontology.
Other points that will be taken into account are the analysis of the efficiency of these systems and their effectiveness on more corpora.

Acknowledgments The work described in this paper is partially supported by grant CR30I1_162758 of the Swiss National Science Foundation.

References

1. Protein ontology (2017), `http://pir.georgetown.edu/pro/pro.shtml`
2. Aiolli, F., Donini, M.: EasyMKL: a scalable multiple kernel learning algorithm. Neurocomputing 169, 215–224 (2015)
3. Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner, W.A., Cohen, K.B., Verspoor, K., Blake, J.A., et al.: Concept annotation in the CRAFT corpus. BMC bioinformatics 13(1), 161 (2012)
4. Bard, J., Rhee, S.Y., Ashburner, M.: An ontology for cell types. Genome biology 6(2), R21 (2005)
5. Basaldella, M., Furrer, L., Tasso, C., Rinaldi, F.: Entity recognition in the biomedical domain using a hybrid approach. Journal of biomedical semantics 8(1), 51 (2017)
6. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 35(8), 1798–1828 (2013)
7. Botstein, D., Cherry, J.M., Ashburner, M., Ball, C., Blake, J., Butler, H., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al.: Gene ontology: tool for the unification of biology. Nat genet 25(1), 25–9 (2000)
8. Campos, D., Matos, S., Oliveira, J.L.: Biomedical named entity recognition: a survey of machine-learning tools. In: Theory and Applications for Advanced Text Mining. InTech (2012)
9. Chiu, B., Crichton, G., Korhonen, A., Pyysalo, S.: How to train good word embeddings for biomedical NLP. In: Proceedings of the 15th Workshop on Biomedical Natural Language Processing. pp. 166–174 (2016)
10. Dang, T.H., Le, H.Q., Nguyen, T.M., Vu, S.T.: D3ner: Biomedical named entity recognition using crf-bilstm improved with fine-tuned embeddings of various linguistic information. Bioinformatics 1, 8 (2018)
11. Das, A., Ganguly, D., Garain, U.: Named entity recognition with word embeddings and wikipedia categories for a low-resource language. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 16(3), 18 (2017)

12. Degtyarenko, K., De Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M., Ashburner, M.: ChEBI: a database and ontology for chemical entities of biological interest. Nucleic acids research 36(suppl_1), D344–D350 (2007)
13. Donini, M., Aiolli, F.: Learning deep kernels in the space of dot product polynomials. Machine Learning 106(9-10), 1245–1269 (2017)
14. Eilbeck, K., Lewis, S.E., Mungall, C.J., Yandell, M., Stein, L., Durbin, R., Ashburner, M.: The sequence ontology: a tool for the unification of genome annotations. Genome biology 6(5), R44 (2005)
15. Federhen, S.: The NCBI taxonomy database. Nucleic acids research 40(D1), D136–D143 (2011)
16. Gönen, M., Alpaydın, E.: Multiple kernel learning algorithms. Journal of machine learning research 12(Jul), 2211–2268 (2011)
17. Habibi, M., Weber, L., Neves, M., Wiegandt, D.L., Leser, U.: Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics 33(14), i37–i48 (2017)
18. Lee, K.J., Hwang, Y.S., Kim, S., Rim, H.C.: Biomedical named entity recognition using two-phase model based on SVMs. Journal of Biomedical Informatics 37(6), 436–447 (2004)
19. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
20. Nadeau, D., Sekine, S.: A survey of named entity recognition and classification. Lingvisticae Investigationes 30(1), 3–26 (2007)
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of machine learning research 12(Oct), 2825–2830 (2011)
22. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010), http://is.muni.cz/publication/884893/en
23. Rinaldi, F., Clematide, S., Marques, H., Ellendorff, T., Romacker, M., Rodriguez-Esteban, R.: Ontogene web services for biomedical text mining. BMC bioinformatics 15(14), S6 (2014)
24. Seok, M., Song, H.J., Park, C.Y., Kim, J.D., Kim, Y.s.: Named entity recognition using word embedding as a feature. Int. J. Softw. Eng. Appl 10(2), 93–104 (2016)
25. Settles, B.: Biomedical named entity recognition using conditional random fields and rich feature sets. In: Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and Its Applications. pp. 104–107. JNLPBA '04, Association for Computational Linguistics, Stroudsburg, PA, USA (2004), http://dl.acm.org/citation.cfm?id=1567594.1567618