CRFVoter: Chemical Entity Mention, Gene and Protein Related Object recognition using a conglomerate of CRF based tools

Wahed Hemati, Alexander Mehler, and Tolga Uslu

Text Technology Lab,
Goethe Universitt Frankfurt am Main, Germany
{hemati,amehler,uslu}@em.uni-frankfurt.de
http://www.hucompute.org/

Abstract. This paper relates to the two offline *BioCreative V.5 Becalm* tasks. The first challenge is CEMP, the recognition of chemical named entity mentions. The second challenge is GPRO, the recognition of gene and protein related objects in running text. We focus on training and optimizing state-of-the-art solutions for named entity tagging for CEMP and GPRO. Finally, we present `CRFVoter`, a two staged application of CRF.

Key words: Biocreative, BeCalm,Chemical named entity recognition, Named Entity Recognition, CRF

1 Introduction

BioCreative V.5 consists of two offline tasks, namely CEMP (Chemical Entity Mention Recognition) and GPRO (Gene and Protein Related Object Recognition). CEMP requires the detection of chemical named entity mentions. The task requires detecting correctly the start and end indices corresponding to chemical entities. GPRO task requires identifying mentions of gene and protein related objects mentioned in patent titles and abstracts. In this work, we survey Named Entity Recognition techniques, which is an abstraction of the CEMP and GPRO task. Our survey includes 5 state-of-the-art NER systems and two combination techniques for these systems, namely Majority vote and `CRFVoter`.

2 Corpus

The organizers of *BioCreative V.5* provided a corpus of 21000 patent abstracts (titles and abstracts in English) from patents published between 2005 and 2014. The corpus is manually annotated for the CEMP and GPRO tasks. For our experiments we divided the corpus in 60 % training set, 25 % development set and 15 % test set by means of random sampling. We applied multiple preprocessing steps on each set including sentence splitting, tokenization, lemmatization and fine-grained morphological tagging. These and related information units were used as features in our experiments.

Table 1. Differences of labeled output between each pair of NER system.

	Stanford	**MarMoT**	**CRF++**	**MITIE**	**Glample**
Stanford	0	2,29 %	2,12 %	2,44 %	2,50 %
MarMoT		0	2,56 %	2,61 %	2,43 %
CRF++			0	2,91 %	2,47 %
MITIE				0	2,51 %
Glample					0

3 System Description

In this section, we present a survey of Named Entity Recognizer trained for the CEMP and GPRO tasks. For each NER we optimized the hyperparameter settings. Hyperparameter tuning is a challenging topic in *Machine Learning* (ML). The optimal set of hyperparameters depends on the model, dataset and the domain. To this end, we focused in our experiments on optimizing hyperparameter, which lead to a noticeable increase of F-score compared to default settings. For each NER, we used grid search on a set of configurations of hyperparameter and trained them accordingly, choosing the hyperparameter configuration that gives the best performance. The big downside of optimizing hyperparameter is to overfit the model on training data. Each NER classifies a different subset correctly. Table 1 shows the pairwise differences between NER systems. Therefore, a combination of these NER was seemingly promising in order to increase precision and recall, due to possible orthogonal output labels. To this end, we experimented with a simple majority vote. Further more, we developed a two-stage application of CRF for combinations of sequence labeling tools, called `CRFVoter`. We trained each NER system on the training set and tested against the test set (see Section 2). In this work, we consider the NER systems as enumerated in Table 1 and described in the following subsections.

3.1 Stanford Named Entity Recognizer

Stanford Named Entity Recognizer[1] is a Java implementation of a CRF based Named Entity Recognizer [1]. Table 2 shows the hyperparameter space used in our experiments. The combination of parameters results in 432 model files. The best performing set of features for GPRO, marked with ♣, leads to an F-score of 0,82. The worst setting results in an F-score of 0,73. The best performing feature set for CEMP is marked with ♢ and produces an F-score of 0,825; the worst setting results in 0,74.

[1] `http://nlp.stanford.edu/software/CRF-NER.shtml`

Table 2. Parameter Space of Stanford Named Entity Recognizer.

Parameter	Possible Values
Use n-Grams	[true$_{\clubsuit,\diamondsuit}$; false]
No mid-n-Grams	[true$_{\clubsuit}$; false$_{\diamondsuit}$]
Use Disjunctive	[true$_{\clubsuit,\diamondsuit}$, false]
Use Type Sequences	[true$_{\clubsuit,\diamondsuit}$, false]
Max Left	[1$_{\diamondsuit}$,2$_{\clubsuit}$,3]
Max Right	[1,2,3$_{\clubsuit,\diamondsuit}$]
Max N-Gram Length	[2,4,6$_{\clubsuit,\diamondsuit}$]
Combination count: 432	

3.2 MarMoT

MarMoT[2] is a generic CRF framework [2]. MarMoT implements a higher order CRF with approximations such that it can deal with large output spaces. Additionally it can be trained to fire on the predictions of lexical resources (gazette files) and on word embeddings [2]. Table 3 shows the hyperparameter space used in our experiments for MarMoT. The combination of parameters results in 3888 model files. The best performing set of features for GPRO is marked with ♣ and produces an F-score of 0,72. The worst set results in an F-score of 0,59. The best performing set of features for CEMP is marked with ♢ and generates an F-score of 0,85. The worst set results in a F-score of 0,61.

Table 3. Parameter Space of MarMoT.

Parameter	Possible Values
Num iterations	[10,20$_{\clubsuit,\diamondsuit}$]
Penalty	[0$_{\clubsuit,\diamondsuit}$,1,2]
Beam size	[1$_{\clubsuit}$,2$_{\diamondsuit}$,5]
Quadratic penalty	[0$_{\clubsuit,\diamondsuit}$,1,2]
Order	[1$_{\clubsuit,\diamondsuit}$,2,3,4]
Prob threshold	[0.01,0.001$_{\clubsuit,\diamondsuit}$]
Effective order	[1$_{\clubsuit,\diamondsuit}$,2,3]
Num chunks	[2$_{\clubsuit,\diamondsuit}$,5,10]
Combination count: 3888	

3.3 CRF++

CRF++[3] is a customizable open source implementation of CRF [3]. In our experiments we used unigram and bigram features, containing the current, previous

[2] `http://cistern.cis.lmu.de/marmot/`
[3] `http://taku910.github.io/crfpp/`

and the next word. Table 4 shows the hyperparameter space used in our experiments for CRF++. The combination of parameters results in 20 model files. The best performing set of parameters for GPRO is marked with ♣ and generates an F-score of 0,69. The worst set results in an F-score of 0,04. The best performing set of parameters for CEMP is marked with ♢ producing an F-score of 0,73, while the worst setting results in an F-score of 0,42.

Table 4. Parameter Space of MarMoT

Parameter	**Possible Values**
c	[0.6, 1, 1.6, 3, 5, 7, 15♣, 50, 100, 1000]
a	[CRF-L1, CRF-L2♣]
Combination count: 20	

3.4 MITIE

MITIE is a open source information extraction tool. MITIE can be trained using techniques like distributional word embeddings and *Structural Support Vector Machines* [4]. Due to the lack of documentation, we did not optimize MITIE. The default configuration for named entity recognition produces an F-score of 0,65 for GPRO and 0,62 for CEMP.

3.5 Glample NER Tagger

Glample NER Tagger is a neural-network-based named entity recognizer. It is based on Bidirectional LSTM and CRF[5]. Due to the long-lasting training time, only the default parameter settings were considered. This resulted in an F-score of 0,75 for GPRO and 0,77 for CEMP.

3.6 Majority Vote

By means of majority voting, we combined the best performing outputs of each of the NER systems considered so far. We selected the label that was most frequently output by the different NER systems. Majority voting reaches an F-score of 0,71 for GPRO, which is below the best performing system considered so far. For CEMP majority voting results in an F-score of 0,78. Facing these results we can state that a simple majority vote brings no gain in precision and recall.

3.7 CRFVoter

Since majority voting did not better F-score, we developed the so-called `CRFVoter`, that is, a two-stage CRF-system for combining different sequence labeling systems. In the first stage each NER is optimized independently (see Section 3) on the trainings set. In the second stage, the development set (see Section 2) is tagged by each NER independently. The output label of each NER system is taken as individual feature for `CRFVoter`. Figure 1 exemplifies `CRFVoter` on the input stream

"*Inhibitors of D-amino acid oxidase ...*"

For each token of this stream, the corresponding labels are calculated by the NER systems of Section 3. In the second stage, the output labels of each NER system are taken as individual features for a CRF operating on the latter system's output. The `CRFVoter` trains a model based on these features. For tagging, `CRFVoter` takes again the output of each NER system as features and labels the sequence by means of the 2nd-stage CRF. In the example of Figure 1, majority voting would tag the sequence wrongly. On the other hand, `CRFVoter` learned the correct sequence of labels. `CRFVoter` achieved an F-score of 0,84 on GPRO and 0,88 on CEMP – both outcomes are better then any of the best performing NER systems documented in Section 3.

4 Results

Table 5 shows the comparison of annotators trained for GPRO and Table 6 considers the corresponding results with respect to CEMP. The best performing annotator is `CRFVoter` in both tasks when being tested on the test set described in Section 2. On the blinded test set for CEMP provided by the Biocreative team, the best performing system, which is again `CRFVoter`, reaches an F-score of **0,87**. For the blinded test set provided for GPRO, it reaches an F-score of **0,75**.

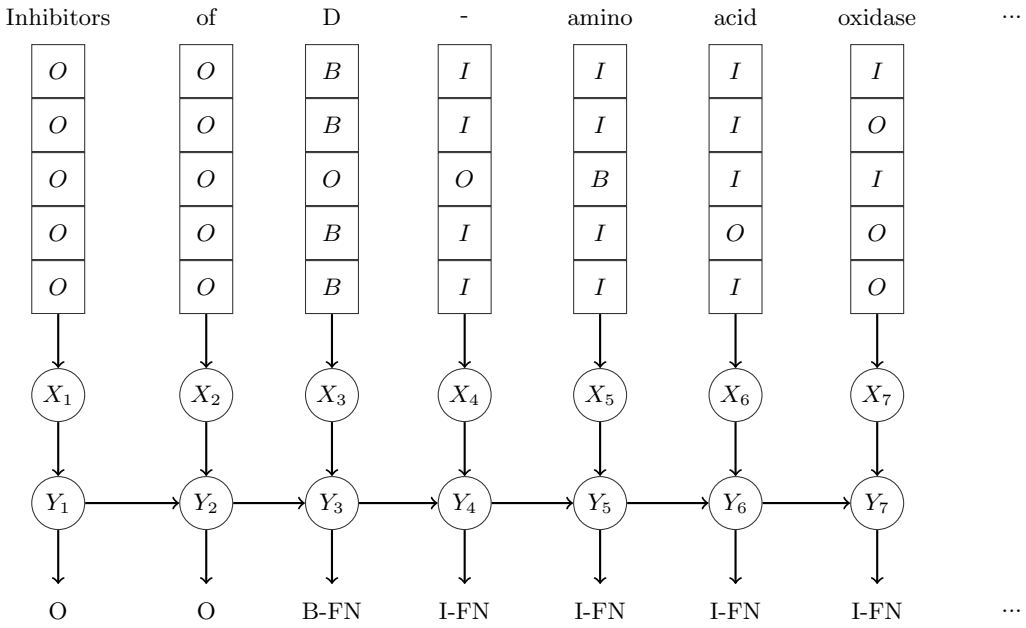
Table 5. Comparison of annotators trained for GPRO.

System	P	R	F
Stanford NER	0,83	0,82	**0,82**
MarMoT	0,76	0,69	0,72
CRF++	0,75	0,64	0,69
MITIE	0,74	0,58	0,65
Glample	0,79	0,72	0,75
Majority Vote	0,72	0,71	0,72
`CRFVoter`	0,85	0,84	**0,84**

Table 6. Comparison of annotators trained for CEMP.

System	P	R	F
Stanford NER	0,85	0,80	0,82
MarMoT	0,85	0,85	**0,85**
CRF++	0,77	0,73	0,73
MITIE	0,62	0,61	0,62
Glample	0,76	0,79	0,77
Majority Vote	0,78	0,79	0,78
`CRFVoter`	0,88	0,87	**0,88**

Fig. 1. Architecture of `CRFVoter` exemplified by means of a single sentence.



5 Discussion and future work

In this work, we compared a set of NER systems. We trained and optimized every NER system for GPRO and CEMP by means of hyperparameter optimization. We showed that optimizing hyperparameter can be crucial. One NER system in our experiments gained an improvement of more then 60 %. In future work, a hyperparameter search algorithm, which is less time-consuming than grid search, will be implemented, for instance, *random search* or *Bayesian optimization*. A bigger hyperparameter space can then be searched and also non-trivial values for continuous variables can be optimized. We additionally introduced and evaluated the so-called `CRFVoter`, a two-stage CRF tool for combining underlying sequence modeling tools (as given by the NER of our comparative study). `CRFVoter` gained 2 % improvement compared to the best performing reference systems being examined in our study. Thus, `CRFVoter` may be further-developed by feeding it with the output of additional sequence labeling systems. This will be the second part of our future work.

6 Acknowledgment

We gratefully acknowledge support by the *Deutsche Forschungsgemeinschaft* via the Specialised Information Services *Biodiversity Research*.

References

1. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 363–370
2. Mueller, T., Schmid, H., Schütze, H.: Efficient higher-order CRFs for morphological tagging. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, Association for Computational Linguistics (October 2013) 322–332
3. Kudo, T.: CRF++: Yet another CRF toolkit. Software available at https://taku910.github.io/crfpp/ (2005)
4. Geyer, K., Greenfield, K., Mensch, A., Simek, O.: Named entity recognition in 140 characters or less. In: Microposts. (2016)
5. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. CoRR (2016)

View publication stats