Exploring Deep Knowledge Resources in Biomedical Name Recognition

ZHOU GuoDong   SU Jian
Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
Email: {zhougd, sujian}@i2r.a-star.edu.sg

Abstract

In this paper, we present a named entity recognition system in the biomedical domain. In order to deal with the special phenomena in the biomedical domain, various evidential features are proposed and integrated through a Hidden Markov Model (HMM). In addition, a Support Vector Machine (SVM) plus sigmoid is proposed to resolve the data sparseness problem in our system. Besides the widely used lexical-level features, such as word formation pattern, morphological pattern, out-domain POS and semantic trigger, we also explore the name alias phenomenon, the cascaded entity name phenomenon, the use of both a closed dictionary from the training corpus and an open dictionary from the database term list SwissProt and the alias list LocusLink, the abbreviation resolution and in-domain POS using the GENIA corpus.

1. The Baseline System

1.1 Hidden Markov Model

In this paper, we use the Hidden Markov Model (HMM) as described in Zhou et al (2002). Given an output sequence $O_1^n = o_1 o_2 ... o_n$, the system finds the most likely state sequence $S_1^n = s_1 s_2 ... s_n$ that maximizes $P(S_1^n \mid O_1^n)$ as follows:

$$\log P(S_1^n \mid O_1^n) = \log P(S_1^n) - \sum_{i=1}^{n} \log P(s_i) + \sum_{i=1}^{n} \log P(s_i \mid O_1^n) \qquad (1)$$

From Equation (1), we can see that:

- The first term can be computed by applying chain rules. In ngram modeling (Chen et al 1996), each tag is assumed to be dependent on the N-1 previous tags.
- The second term is the summation of log probabilities of all the individual tags.
- The third term corresponds to the "lexical" component (dictionary) of the tagger.

The idea behind the model is that it tries to assign each output an appropriate tag (state), which contains boundary and class information. For example, "*TCF 1 binds stronger than NF kB to TCEd DNA*". The tag assigned to token "*TCF*" should indicate that it is at the beginning of an entity name and it belongs to the "*Protein*" class; and the tag assigned to token "*binds*" should indicate that it does not belong to an entity name. Here, the Viterbi algorithm (Viterbi 1967) is implemented to find the most likely tag sequence.

The problem with the above HMM lies in the data sparseness problem raised by $P(s_i \mid O_1^n)$ in the third term of Equation (1). In this paper, a Support Vector Machine (SVM) plus sigmoid is proposed to resolve this problem in our system.

1.2 Support Vector Machine plus Sigmoid

Support Vector Machines (SVMs) are a popular machine learning approach first presented by Vapnik (1995). Based on the structural risk minimization of statistical learning theory, SVMs seek an optimal separating hyper-plane to divide the training examples into two classes and make decisions based on support vectors which are selected as the only effective examples in the training set. However, SVMs produce an un-calibrated value that is not probability. That is, the unthresholded output of an SVM can be represented as

$$f(x) = \sum_{i \in SV} a_i \cdot y_i \cdot k(x_i, x) + b \qquad (2)$$

To map the SVM output into the probability, we train an additional sigmoid model(Platt 1999):

$$p(y = 1 \mid f) = \frac{1}{1 + \exp(Af + B)} \qquad (3)$$

Basically, SVMs are binary classifiers. Therefore, we must extend SVMs to multi-class (e.g. K) classifiers. For efficiency, we apply the *one vs. others* strategy, which builds K classifiers so as to separate one class from all others, instead of the *pairwise* strategy, which builds K*(K-1)/2 classifiers considering all pairs of classes. Moreover, we only apply the simple linear kernel, although other kernels (e.g. polynomial kernel) and pairwise strategy can have better performance.

1.3 Features

Various widely used lexical-level features are explored in the baseline system.

- **Word Formation Pattern (F_{WFP}):** The purpose of this feature is to capture capitalization, digitalization and other word formation information. In this paper, the same feature as in Shen et al 2003 is used.
- **Morphological Pattern (F_{MP})**: Morphological information, such as prefix and suffix, is considered as an important cue for terminology identification. Same as Shen et al 2003, we use a statistical method to get the most useful prefixes/suffixes from the training data.
- **Part-of-Speech (F_{POS}):** Since many of the words in biomedical entity names are in lowercase, capitalization information in the biomedical domain is not as evidential as that in the newswire domain. Moreover, many biomedical entity names are descriptive and very long. Therefore, POS may provide useful evidence about the boundaries of biomedical entity names. In the baseline system, an out-domain POS using the PENN TreeBank is applied.
- **Head Noun Trigger (F_{HEAD}):** The head noun, which is the major noun of a noun phrase, often describes the function or the property of the noun phrase. In this paper, we automatically extract unigram and bigram head nouns from the training data, and rank them by frequency. For each entity class, we select 50% of top ranked head nouns as head noun triggers.

2. Deep Knowledge Resources

Besides the widely used lexical-level features as described above, we also explore the name alias phenomenon, the cascaded entity name phenomenon, the use of both a closed dictionary from the training corpus and an open dictionary from the database term list SwissProt and the alias list LocusLink, the abbreviation resolution and in-domain POS using the GENIA corpus.

2.1 Name Alias Resolution

A novel name alias feature is proposed to resolve the name alias phenomenon. The intuition behind this feature is the name alias phenomenon that relevant entities will be referred to in many ways throughout a given text and thus success of named entity recognition is conditional on success at determining when one noun phrase refers to the very same entity as another noun phrase.

During decoding, the entity names already recognized from the previous sentences of the document are stored in a list. When the system encounters an entity name candidate (e.g. a word with a special word formation pattern), a name alias algorithm (similar to Schwartz et al 2003) is invoked to first dynamically determine whether the entity name candidate might be alias for a previously recognized name in the recognized list. The name alias feature F_{ALIAS} is represented as *ENTITYnLm* (L indicates the locality of the name alias phenomenon). Here *ENTITY* indicates the class of the recognized entity name and *n* indicates the number of the words in the recognized entity name while *m* indicates the number of the words in the recognized entity name from which the name alias candidate is formed. For example, when the decoding process encounters the word "*TCF*", the word "*TCF*" is proposed as an entity name candidate and the name alias algorithm is invoked to check if the word "*TCF*" is an alias of a recognized named entity. If "*T cell Factor*" is a "*Protein*" name recognized earlier in the document, the word "*TCF*" is determined as an alias of "*T cell Factor*" with the name alias feature *Protein3L3* by taking the three initial letters of the three-word "*protein*" name "*T cell Factor*".

2.2 Cascaded Entity Name Resolution

It is found (Shen et al 2003) that 16.57% of entity names in GENIA V3.0 have cascaded constructions, e.g.

<RNA><DNA>CIITA</DNA> mRNA</RNA>.

Therefore, it is important to resolve such phenomenon.

Here, a pattern-based module is proposed to resolve the cascaded entity names while the above HMM is applied to recognize embedded entity names and non-cascaded entity names. In the GENIA corpus, we find that there are six useful patterns of cascaded entity name constructions:

- <ENTITY> := <ENTITY> + head noun, e.g. <PROTEIN> binding motif→<DNA>
- <ENTITY> := <ENTITY> + <ENTITY>
- <ENTITY> := modifier + <ENTITY>, e.g. anti <Protein>→<Protein>
- <ENTITY> := <ENTITY> + word + <ENTITY>
- <ENTITY> := modifier + <ENTITY> + head noun
- <ENTITY> := <ENTITY> + <ENTITY> + head noun

In our experiments, all the rules of above six patterns are extracted from the cascaded entity names in the GENIA V3.0 to deal with the cascaded entity name phenomenon where the <ENTITY> above is restricted to the five categories in the shared task: Protein, DNA, RNA, CellLine, CellType.

2.3 Abbreviation Resolution

While the name alias feature is useful to detect the inter-sentential name alias phenomenon, it is unable to identify the inner-sentential name alias phenomenon: the inner-sentential abbreviation. Such abbreviations widely occur in the biomedical domain.

In our system, we present an effective and efficient algorithm to recognize the inner-sentential abbreviations more accurately by mapping them to their full expanded forms. In the GENIA corpus, we observe that the expanded form and its abbreviation often occur together via parentheses. Generally, there are two patterns: “expanded form (abbreviation)” and “abbreviation (expanded form)”.

Our algorithm is based on the fact that it is much harder to classify an abbreviation than its expanded form. Generally, the expanded form is more evidential than its abbreviation to determine its class. The algorithm works as follows: Given a sentence with parentheses, we use a similar algorithm as in Schwartz et al (2003) to determine whether it is an abbreviation with parentheses. If yes, we remove the abbreviation and the parentheses from the sentence. After the sentence is processed, we restore the abbreviation with parentheses to its original position in the sentence. Then, the abbreviation is classified as the same class of the expanded form, if the expanded form is recognized as an entity name. In the meanwhile, we also adjust the boundaries of the expanded form according to the abbreviation, if necessary. Finally, the expanded form and its abbreviation are stored in the recognized list of biomedical entity names from the document to help the resolution of forthcoming occurrences of the same abbreviation in the document.

2.4 Dictionary

In our system, two different features are explored to capture the existence of an entity name in a closed dictionary and an open dictionary. Here, the closed dictionary is constructed by extracting all entity names from the training data while the open dictionary (~700,000 entries) is combined from the database term list Swissport and the alias list LocusLink. The closed dictionary feature is represented as Closed*ENTITYn* (Here *ENTITY* indicates the class of the entity name and *n* indicates the number of the words in the entity name) while the open dictionary feature is represented as Open*n* (Here *n* indicates the number of the words in the entity name. We don’t differentiate the class of the entity name since the open dictionary only contains protein/gene names and their aliases).

2.5 In-domain POS

We also examine the impact of an in-domain POS feature instead of an out-domain POS feature which is trained on PENN TreeBank. Here, the in-domain POS is trained on the GENIA corpus V3.02p.

3. Evaluation

Table 1 shows the performance of the baseline system and the impact of deep knowledge resources while Table 2-4 show the detailed performance using the provided scoring algorithm. Table 1 shows that:

- The baseline system achieves F-measure of 60.3 while incorporation of deep knowledge resources can improve the performance by 12.2 to 72.5 in F-measure.
- The replacement of the out-domain POS with in-domain POS improves the performance by 3.8 in F-measure. This suggests in-domain POS can much improve the performance.
- The name alias feature in name alias resolution slightly improves the performance by 0.9 in F-measure.
- The cascaded entity name resolution improves the performance by 3.1 in F-measure. This suggests that the cascaded entity name resolution is very useful due to the fact that about 16% of entity names have cascaded constructions.
- The abbreviation resolution improves the performance by 2.1 in F-measure.
- The small closed dictionary improves the performance by 1.5 in F-measure. In the meanwhile, the large open dictionary improves the performance by 1.2 in F-measure largely due to the performance improvement for the protein class. It is interesting that the small closed dictionary contributes more than the large open dictionary does. This may be due to the high ambiguity in the open dictionary and that the open dictionary only contains protein and gene names.

Table 1: Impact of Deep Knowledge Resources

Performance	F
Baseline	60.3

+In-domain POS	+3.8
+Name Alias Feature	+0.9
+Cascaded Entity Name Res.	+3.1
+Abbreviation Resolution	+2.1
+Small Closed Dictionary	+1.5
+Large Open Dictionary	+1.2
+All Deep Knowledge Resources	**+12.2**

Table 2: Final Detailed Performance: full correct answer

(# of correct answers)	P	R	F
Protein (4015)	69.01	79.24	73.77
DNA (772)	66.84	73.11	69.83
RNA (75)	64.66	63.56	64.10
Cell Line (329)	53.85	65.80	59.23
Cell Type (1391)	78.06	72.41	75.13
Overall (6582)	**69.42**	**75.99**	**72.55**

Table 3: Final Detailed Performance: correct left boundary with correct class information

(# of correct answers)	P	R	F
Protein (4239)	72.86	83.66	77.89
DNA (798)	69.09	75.57	72.18
RNA (76)	65.52	64.41	64.96
Cell Line (346)	56.63	69.20	62.29
Cell Type (1418)	79.57	73.82	76.59
Overall (6877)	**72.53**	**79.39**	**75.80**

Table 4: Final Detailed Performance: correct right boundary with correct class information

(# of correct answers)	P	R	F
Protein (4285)	73.65	84.57	78.73
DNA (854)	73.94	80.87	77.25
RNA (83)	71.55	70.34	70.94
Cell Line (383)	62.68	76.60	68.95
Cell Type (1532)	85.97	79.75	82.74
Overall (7137)	**75.27**	**82.39**	**78.67**

4. Conclusion

In the paper, we have explored various deep knowledge resources such as the name alias phenomenon, the cascaded entity name phenomenon, the use of both a closed dictionary from the training corpus and an open dictionary from the database term list SwissProt and the alias list LocusLink, the abbreviation resolution and in-domain POS using the GENIA corpus.

In the near future, we will further improve the performance by investigating more on conjunction and disjunction construction and the combination of coreference resolution.

Acknowledgement

We thank ZHANG Zhuo for providing the database entity name list SwissProt and the alias list LocusLink.

References

Chen and Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of the 34th Annual Meeting of the Association of Computational Linguistics (ACL'1996)*. pp310-318. Santa Cruz, California, USA.

Ohta T., Tateisi Y., Kim J., Mima H., and Tsujii J. 2002. The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In *Proc. of HLT 2002*.

Platt J. 1999. Probabilistic Outputs for Support Vector Machines and comparisions to regularized Likelihood Methods. *MIT Press.*

Schwartz A.S. and Hearst M.A. 2003. A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text. In *Proc. of the Pacific Symposium on Biocomputing (PSB 2003)* Kauai.

Shen Dan, Zhang Jie, Zhou GuoDong, Su Jian and Tan Chew Lim, Effective Adaptation of a Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain, *Proceedings of ACL'2003 Workshop on Natural Language Processing in Biomedicine*, Sapporo, Japan, 11 July 2003. pp49-56.

Vapnik V. 1995. *The Nature of Statistical Learning Theory*. NY, USA: Springer-Verlag.

Viterbi A.J. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 260-269.

Zhou G.D. and Su J. 2002. Named Entity Recognition using an HMM-based Chunk Tagger. In *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 473-480.