BioTagger: A Biological Entity Tagging System

Hongfang Liu, Department of Information Systems, UMBC
(hfliu@umbc.edu)
Cathy Wu, PIR, Georgetown University
(wuc@georgetown.edu)
Carol Friedman, Department of Biomedical Informatics, Columbia University
(friedman@dbmi.columbia.edu)

System Description

The results submitted for Task 1B is the outcome of the prototype system of an ongoing biological entity tagging system, called BioTagger, which decomposes the tagging task into several subtasks and considers novelty, synonymy and ambiguity associated with terms representing biological entities in text:

- Automatic construction of a comprehensive dictionary for biological entities using online resources
- Automatic acquisition of disambiguation knowledge from these resources
- Intelligent dictionary lookup that considers novelty, synonymy, and ambiguity
- Training a POS tagger using unsupervised machine learning techniques to further consider novelty and ambiguity, and training disambiguation classifiers to perform corpus-based disambiguation to further resolve the ambiguity

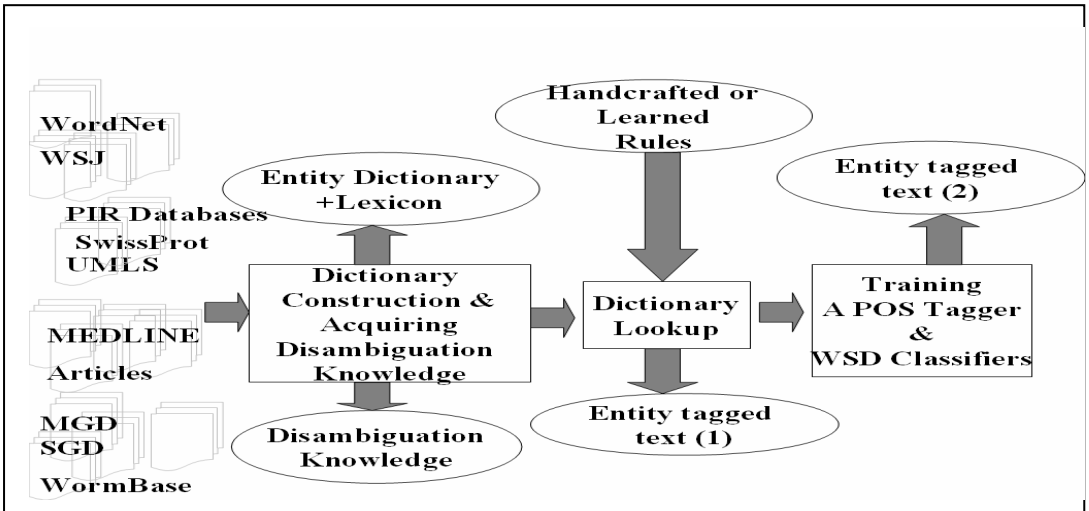

Figure1. Overview of the architecture

Figure 1 shows the overall architecture of the system. We first derive a comprehensive dictionary using a large collection of online resources. At the same time, these resources are utilized to acquire disambiguation knowledge for each entity such as definitions, cross-reference, or co-occurrence information. An intelligent dictionary lookup tool is then constructed that identifies novelty using handcrafted or learned rules, recognizes synonymy, and performs disambiguation using previous acquired knowledge. Machine learning techniques are then implemented for the training of WSD classifiers for frequently occurring ambiguous terms and a POS tagger. The first three tasks have been implemented in the prototype system. In the following, we show the details of each of them tailoring to the BioCreative.

Automatic construction of a biological entity dictionary – For each organism entity, we collected terms to represent the entity by combining each model organism database entry with Swiss-prot or Trembl databases using the cross-reference information supplied by model organism databases (e.g., MRK_Swissprot.rpt.txt). Note that the parenthetical expressions in the Swiss-Prot and Trembl were separated into several terms. For example, we had four terms (i.e., *Pigment epithelium-derived factor precursor*, *PEDF*, *Stromal cell-derived factor 3*, and *SDF-3*) extracted from the description field for protein *PEDF_MOUSE*: *Pigment epithelium-derived factor precursor (PEDF) (Stromal cell-derived factor 3) (SDF-3)).*

Automatic acquisition of disambiguation knowledge - We extracted all words from terms for each entity. After removing 765 stopping words (e.g., "this", "gene", or "fragment") obtained by combining the general English word list in mwords [1] with the top 2000 frequency words in MEDLINE, all remaining words are considered as a disambiguation vector for the corresponding entity.

Intelligent dictionary lookup – We applied a pattern-matching method developed previously [2] to recognize definitions for defined abbreviations in parenthetical expressions. We also handled textual variants caused by punctuation marks, lexical variants, and synonyms. Each term in the dictionary was tokenized into a vector consisting of consecutive letters, numbers, and punctuation marks. Then the vector was normalized sequentially according to the following steps while the output of each step was recorded and would be used for dictionary lookup:

- Exact – tokens are concatenated using "+".
- Lower case -- normalized every token into lower case in the token vector and concatenated using "+". For example, the normalized string for ("M", "5", "R") is m+5+r.
- Ignoring punctuation – tokens that are punctuation marks are ignored and the remaining tokens are concatenated using "+". For example, the normalized string in this step for ("cholinergic", " ", "receptor", ",", "muscarinic", " ", "5") is cholinergic+receptor+muscarinic+5.
- Sorted – tokens are sorted and concatenated using "+". For example, the normalized string in this step for (*"cholinergic"," ","receptor",",","muscarinic"," ","5"*) is *5+cholinergic+muscarinic+receptor.*
- Synonym-like-replacement-- We applied a synonym-like set for each organism, which contains pairs (w1, w2), where w1 and w2 are only different words in two names of the same entity disregarding word orders. For example, (*cholinergic*, *acetylcholine*) is a synonym-like pair, which is derived from entries of *MGI:109248, cholinergic receptor, muscarinic 5* and *muscarinic acetylcholine receptor 5*.

All entities associated with the same exact or normalized string are concatenated and strings with multiple entities are ambiguous. For example, *php+-+2* corresponds to two identifiers: *MGI:1932286 (Hypoxia-inducible factor prolylhydroxylase 2)* and *MGI:96215 (hyperphenylalaninemia 2).* Additionally, we compiled a family entity dictionary using terms that are only different at the right part of the terms with the numbers or Greek letters. For example, an entry in the family dictionary for mouse is *looptail* (derived from *MGI:2671533 looptail 2* and *MGI:2671536 looptail 3*) and is associated with two entities (i.e., *MGI:2671533* and *MGI: 2671536*).

For abstracts seeking dictionary lookup, we first tokenized the whole text into sentences. Each sentence was then tokenized to tokens. For each sentence, we checked parenthetical expressions to see if they were used to define an abbreviation. If yes, a local synonym pair *(abbreviation, full name)* was defined and it will be used for disambiguation and for synonym-like-replacement mapping. The mapping used the longest string matching first method. It began with the string consisting of the first 10 tokens in the sentence and subsequently shrinked the size of the tokens from the right side when no matching was found. Each string was normalized using the normalization method applied for dictionary entries until the size of the token reached 0 or a mapping was found. If the string was a capitalized string followed by the lower case "s" or the string was terminated with words *proteins*, *enzymes*, *genes*, *families*, *mRNAs*, *transporters*, *receptors*, *homologs* etc, we tried to find mapping using the family dictionary. If the string contained specialized patterns which usually were abbreviated forms for several entities from the same family (e.g., *HAP2,3,4* or *HAP2-4*, *HAP-2, -3, and -4*, or *HAP2/4*), we separated them and reassembled to several strings and tried to find mapping for each of them. For example, *HAP2/4* would become two strings *HAP2* and *HAP4*. For terms representing multiple entities, we computed a similarity measure (NCW * log (NW +1) / NW) between the abstract and the disambiguation vector we acquired, where NCW is the number of common words in the abstract and the disambiguation vector, and NW is the total number of words in the abstract and the disambiguation vector. The term was tagged with the entity associated with the highest similarity measure which at the same time must be over a threshold.

Final Submission: For mouse and yeast, we managed to submit three runs with different threshold settings. The threshold settings here mostly are thresholds for disambiguation of a gene or protein names from general English words. For fly, we submitted only one run because of the high ambiguous nature of biological entities in Flybase itself. There are a tremendous amount of systematic ambiguous entities or conceptual-related ambiguous strings in the constructed dictionary for Flybase. Such ambiguity is unavoidable which can be evidenced by the second identifiers (ID2) in Flybase database. For example, in the current Flybase, a lot of entities share the same second identifiers. Additionally, Flybase includes data on all species from the family Drosophilidae even primary species represented is Drosophia melanogaster. Even the guideline indicates we only need to tag Drosophia melanogaster genes and their products, however, in the training set, there are genes or gene products from other species that are tagged. So we ended up with a high ambiguous entity dictionary for Flybase. We managed to submit one run for Flybase using a very strict disambiguation threshold setting: for each identified entity in an abstract, the disambiguation threshold for that entity must over a very high threshold. So the result may miss a lot of entities, and at the same time, may choose the wrong entities for those ambiguous strings since the dictionary we acquired includes all species from the family Drosophilidae.

Result

Table 1 shows our result. We will not give detail analysis of fly since entities we constructed contain all Drosophilidae species which was not quite right in the first place. Our system has the best recall for both mouse and yeast. However, the system has the worst precision at the same time. After communicating with the organization

Organism	F-measure	Precision	Recall	TruePositives	FalsePositives	Missed
Yeast Block	0.773	0.646	0.962	590	324	23
	0.77	0.642	0.962	590	329	23
	0.763	0.661	0.902	553	284	60
Mouse Block	0.582	0.431	0.897	488	645	56
	0.573	0.421	0.897	488	671	56
	0.609	0.492	0.798	434	448	110
Fly Block	0.284	0.224	0.389	167	580	262

Table 1. Highlight our result in among all submissions.

committee, we found that we misunderstood the annotation guideline on handling family genes besides we misunderstood the guideline for tagging fly abstracts. For example, in the sentence of "Homologous genes to so, denoted SIX genes, have been found in vertebrates.", our system tagged every SIX gene(SIX1, SIX2, SIX3, SIX4, and SIX5) while the gold standard set only lists SIX3 since SIX3 was explicitly mentioned in the abstract. After ignoring false positive caused by tagging family names, the system has a precision of 51.4% (comparing to 43.1%) with an F-measure of 64.5% for mouse and a precision of 73.1% (comparing to 64.6%) with an F-measure of 83.1% for yeast. The other cause of low precision is that the dictionary we constructed contains a lot of terms that are not valid gene or protein names, mostly caused by the inclusion of descriptions in various resources as terms representing the associated entities. Additionally, the gold standard set misses a lot of positive hits which were found by our system and identified by domain experts.

We identified several problems which we plan to investigate further in our future work. One problem is the treatment of the parenthetical expressions in the description fields in various databases. Parentheses in the majority of these descriptions separate terms where each of them can be used to represent the associated protein entity (e.g., the description field for protein *PEDF_MOUSE*: *Pigment epithelium-derived factor precursor (PEDF) (Stromal cell-derived factor 3) (SDF-3))*. However, the parenthetical expressions for the following description for *RL8A_YEAST: 60S ribosomal protein L8-A (L7A-2) (L4-2) (YL5) (RP6)* seem to hold different meanings. Another problem is the use of Trembl database where entries have not been manually curated. For example, our method considers *Chondrocytes* as a term for *MGI:1930004* based on the following description in Trembl: *DD72 protein (Similar to cystatin 10) (Chondrocytes).* Another problem is that in some databases, free text in one field cannot be treated uniquely. For example, in Table 2, terms in some fields (e.g., Locus name, other name, ORF name) in SGD can be used to represent the associated entity. However, in fields Description, Gene Product and Phenotype, some entries can be considered as terms representing the associated entity such as *ATP dependent metalloprotease* for *AFG3*, but some entries are problematic and can not be used to represent the associated entity such as *similar to the CDC48 gene product* for *AFG2*. Some heuristics will be explored to deal with fields which we cannot treat them uniquely. For example, if a phrase contains *similar to* at the beginning, it may not be appropriate to include it in the dictionary.

Locus name	Other name	Description	Gene Product	Phenotype	ORF name	SGDID
AFG1		ATPase family gene	ATPase family		YEL052W	S0000778
AFG2	DRG1	ATPase family gene	similar to the CDC48 gene product	Null mutant is inviable	YLR397C	S0004389
AFG3	YTA10	ATPase family gene	ATP dependent metalloprotease	nuclear petite phenotype; loss of respiratory* competence	YER017C	S0000819
ECI1		enoyl-CoA isomerase	d3,d2-Enoyl-CoA Isomerase	Null mutant is viable but fails to metabolize unsaturated fatty acids	YLR284C	S0004274

Table 2. Example entries in SGD

Conclusion

The result obtained from the competition is encouraging. We plan to continue our research on biological entity tagging, and we believe that we can eventually have a system that can handle novelty, synonymy, and ambiguity problems.

References

1. mword: http://www.dcs.shef.ac.uk/research/ilash/Moby/mwords.html
2. Liu H, AR Aronson and C Friedman (2002). "A study of abbreviations in MEDLINE abstracts". Proc AMIA Symp. 464-468.