A Joint Model for Normalizing Gene and Organism Mentions in Text

Georgi Georgiev* *georgi.georgiev@ontotext.com*
Preslav Nakov† *nakov@comp.nus.edu.sg*
Kuzman Ganchev‡ *kuzman@cis.upenn.edu*
Deyan Peychev* *deyan.peychev@ontotext.com*
Vassil Momchev* *vassil.momtchev@ontotext.com*

Abstract

The aim of gene mention normalization is to propose an appropriate canonical name, or an identifier from a popular database, for a gene or a gene product mentioned in a given piece of text. The task has attracted a lot of research attention for several organisms under the assumption that both the mention boundaries and the target organism are known. Here we extend the task to also recognizing whether the gene mention is valid and to finding the organism it is from. We solve this extended task using a joint model for gene and organism name normalization which allows for instances from different organisms to share features, thus achieving sizable performance gains with different learning methods: Naïve Bayes, Maximum Entropy, Perceptron and MIRA, as well as averaged versions of the last two. The evaluation results for our joint classifier show F_1 score of over 97%, which proves the potential of the approach.

Keywords

Gene normalization, gene mention tagging, organism recognition, identity resolution.

1 Introduction

Gene mention normalization is one of the emerging tasks in bio-medical text processing along with gene mention tagging, protein-protein interaction, and biomedical event extraction. The objective is to propose an appropriate canonical name, or a unique identifier from a predefined list, for each gene or gene product name mentioned in a given piece of text. Solving this task is important for many practical application, e.g., enriching high precision databases such as the *Protein and Interaction Knowledge Base* (PIKB), part of *LinkedLifeData*[1], or compiling gene-related search indexes for large document collections such as *LifeSKIM*[2] and MEDIE[3].

*Ontotext AD, 135 Tsarigradsko Ch., Sofia 1784, Bulgaria
†Department of Computer Science, National University of Singapore, 13 Computing Drive, Singapore 117417
‡Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA

[1] `http://www.linkedlifedata.com`
[2] `http://lifeskim.sirma.bg`
[3] `http://www-tsujii.is.s.u-tokyo.ac.jp/medie/`

In this work, we focus on the preparation of good training data and on improving the performance of the normalization classifier rather than on building an integrated solution for gene mention normalization.

The remainder of the paper is organized as follows: Section 2 gives an overview of the related work, Section 3 present our method, Section 4 describes the experiments and discusses the results, and Section 5 concludes and suggests directions for future work.

2 Related work

Several approaches have been proposed for gene normalization including classification techniques [5], rule-based systems [7, 19], text matching against dictionaries [2], and different combinations thereof.

Systems for gene mention identification and normalization typically work in three stages: (1) identifying candidate mentions in text, (2) determining the semantic intent of each mention, and (3) normalizing by associating each mention with a unique identifier [15].

For example, Crim et al. [5] first recognize the gene mentions in the text, then match them against a lexicon, and finally filter the wrongly annotated matches using a maximum entropy classifier [1].

The problem we address in this work is closest to *Task 1B* in BioCreAtIvE 2004 [8] and to the *Gene Normalization* task in BioCreAtIvE 2006 [16], which provide lexicons of gene identifiers for a particular organism, e.g., human, yeast, mouse and fly, each of which represents a separate challenge. Given a text document and an organism, the tasks ask that a list be produced containing the identifiers (from the lexicon) of all genes for the target organism that are mentioned in the document. Since the relationship between gene names and identifiers is M:M, disambiguation is a central issue.

Even though reported for individual organisms only, the results from the BioCreAtIvE challenge are quite promising. For yeast, the best F_1 was 0.92 [8]. For mouse and fly, the task was found to be more difficult, probably because of the larger numbers of genes, the higher ambiguity in the gene naming conventions (particularly for fly), and the complexity of mouse gene names; for fly, the best F_1 was 0.82, while for mouse it was 0.79 [8]. For human, the best F_1 was 0.81 [16].

Workshop Biomedical Information Extraction 2009 - Borovets, Bulgaria, pages 14–20

3 Method

Our approach is most similar to that of Crim et al. [5], but there are many differences in the details. First, we do gene mention tagging using a one-best structured version of the Margin-Infused Relaxed Algorithm (MIRA) [4] in order to collect high recall gene mentions. Second, instead of using the conventional strict match, we extensively study different string matching distance metrics. Third, we represent the normalization task as a multi-class classification problem by extending it to multiple organisms (mouse and human). We train a joint statistical classifier that recognizes valid gene identifiers and the corresponding organism at the gene mention level. Finally, we allow the human and mouse examples in the joint model to share features, which yields sizable performance gains.

We try our approach with six classifiers: Naïve Bayes [11], Maximum Entropy [1], Perceptron [20], MIRA [4], and averaged versions of the last two [6].

3.1 One-best structured MIRA

In what follows, x_i will denote the generic input sentence, $Y(x_i)$ will refer to the set of all possible labelings of x_i, and y_i will be the "gold" labeling of x_i. For each pair of a sentence x_i and a labeling $y_i \in Y(x_i)$, we will compute a vector-valued feature representation $f(x_i, y_i)$. Given a weight vector w, the score $w \cdot f(x, y)$ ranks the possible labelings of x; we will denote the top-scoring one as $y_w(x)$. As with hidden Markov models [17], for suitable feature functions, $y_w(x)$ can be computed efficiently using dynamic programming. A linear sequence model is given by a weight vector w. The learning portion of our method requires finding a weight vector w that scores the correct labeling of the training data higher than any incorrect labeling. We used a one-best version of MIRA [3, 13] to choose w. MIRA is an online learning algorithm which updates the weight vector w for each training sentence x_i using the following rule:

$$w_{new} = \arg\min_{w} \|w - w_{old}\| \\ w \cdot f(x_i, y_i) - w \cdot f(x, \hat{y}) \geq L(y_i, \hat{y}) \quad (1)$$

where $L(y_i, \hat{y})$ is a measure of the loss of using $\hat{y}$ instead of y_i, and $\hat{y}$ is a shorthand for $y_{w_{old}}(x_i)$.

In the case of a single constraint, this program has a closed-form solution. The most straightforward and most commonly used loss function is the Hamming loss, which sets the loss of labeling y with respect to the gold labeling $y(x)$ as the number of training examples where the two labelings disagree. Since Hamming loss is not flexible enough, we have separated the misclassified training examples on false positives and false negatives. We defined the *high-recall loss* function to penalize only the false negatives as described in Section 3.2. We implemented one-best MIRA and the corresponding loss functions using an in-house toolkit, Edlin, which provides a general machine learning architecture for linear models and an easy to read framework with implementations of popular machine learning algorithms including Naïve Bayes, Maximum Entropy, Perceptron, one-best MIRA, conditional random fields (CRFs), etc.

3.2 Gene tagging

We experimented with the training and the testing abstracts provided by BioCreAtIvE 2006. We tokenized, sentence split, part-of-speech (POS) tagged and chunked them using maximum entropy models trained on Genia[4] corpora. We subsequently trained several sequence taggers, using the standard BIO encoding [18] and different feature sets.

We started with a CRF tagger [10], which yielded a very low recall (R=73.02%). We further experimented with feature induction [12] and with a second-order CRF, but the recall remained unsatisfactory: 74.72% and 76.64%, respectively. Therefore, we abandoned CRFs altogether and adopted structured MIRA, which allows for transparent training with different loss functions. After a number of experiments, we found that the highest recall is achieved with a loss that uses the number of false negatives, i.e., a larger loss update is made whenever the model fails to discover a gene mention, while discovering a spurious sequence of words would be penalized less severely. We experimented with some popular feature sets used previously in [14] including orthographic, POS, chunk, and presence in a variety of domain-specific lexicons, as well as different conjunctions thereof. As Table 1 shows, the final tagger achieved 83.44% recall.

Predicate Name	Regular Expression
Initial Capital	[A-Z].*
Capital Any	[A-Z].
Initial Capital Alpha	[A-Z][a-z]*
All Capitals	[A-Z]+
All Lower	[a-z]+
Capital Mix	[A-Za-z]+
Has Digit	.*[0-9].*
Single Digit	[0-9]
Double Digit	[0-9][0-9]
Natural Number	[0-9]+
Real Number	[-0-9]+[.,]?[0-9]+
Alpha-Numeric	[A-Za-z0-9]+
Roman	[ivxdlcm]+|[IVXDLCM]+
Has Dash	.*-.*
Initial Dash	-.*
End Dash	.*-
Punctuation	[,.;:?!-+“”]
Multidots	..+
Ends with a Dot	[.$]+.*.
Acronym	[A-Z][A-Z.]*.[A-Z.]*
Lonely Initial	[A-Z].
Single Character	[A-Za-z]
Quote	[“]

Table 1: ***The orthographic predicates used in the structured*** MIRA ***gene tagger.*** *The observation list for each token includes a predicate for each regular expression that matches it.*

3.3 Semantic intent of gene mentions

We addressed gene normalization as a classification problem where, given a gene mention and a candidate gene identifier, a yes/no decision is to be made about whether this is the correct identifier for that mention.

[4] www-tsujii.is.s.u-tokyo.ac.jp/GENIA/home/wiki.cgi

We prepared training data for the classifier as follows. We first created an extended lexicon that combines the lexicons for mouse and human from BioCreAtIvE 2004 and 2006, and we matched against it each chunk that was recognized as a potential gene mention by the gene tagger described in Section 3.2. In the process of matching, we ignored case, punctuation and numbers, and we only used maximal matching strings, i.e., sub-strings of matching strings that linked to the same ID were ignored. At the end of this stage, each gene mention was paired with a corresponding gene identifier (mouse or human), and the pair was marked as either positive or negative. Using these pairs as training data, we built a classifier, which achieved 74% recall and 9.3% precision. In order to boost the recall further, we tried different string similarity metrics within the SIMMETRIC package[5], and we selected the Jaro-Winkler distance, a modification of the Jaro distance [9], that represents a good compromise between specificity and speed. We thus achieved 85% recall and 1% precision.

3.4 Joint organism and gene normalization

While the previous step achieved a very high recall, this was at the expense of precision, which dropped to just 1%. We thus trained a classifier to filter out the bad matches as follows. For each gene mention - gene identifier pair from the previous step, we built a classification example, which includes the gene identifier, the gene name and the words from the local context of the gene mention (two words to the left/right of the mention), and a label: *human-positive* match, *mouse-positive* match or *negative* match. Since we aimed to create a joint classification model for gene normalization of human and mouse gene and gene product names, we represented the label tag of each gene mention subject to classification as a complex tag containing simple labels for the organism and an indication on the validity of the gene identifier. Thus, our model jointly decides whether the match is positive/negative and determines the organism it belongs to.

On training, we considered an example positive for human/mouse if the corresponding gene identifier was included in the list of identifiers for the target abstract and organism, and negative otherwise. Below are shown three examples, each containing a candidate gene match, a context of two tokens on each side of the match, and the corresponding label. In the first example, the match is *calcitonin gene-related peptide*, its left context is *that confers*, its right context is *(CGRP*, and its label is *human-positive*. It is annotated as positive for human, since the abstract annotation and the match annotation agree on the Entrez Gene identifer: 27297. Similarly, the second example is positive for mouse. In the third example, p53 is a valid gene name, but it has been annotated with a wrong mouse identifier, MGI:106202, which is not included in the list of gene identifiers for the target abstract.

[5] http://www.dcs.shef.ac.uk/~sam/stringmetrics.html

`that confers calcitonin gene-related peptide ( CGRP` → *human-positive*

`linkage of CnnI to spontaneous` → *mouse-positive*

`to examine p53 expression during` → *negative*

Using this kind of data, we trained a classifier. We used predicates based on the surface form of the words in the gene mention, on the local context, and on the presence in specialized lexicons:

- the matched phrase (i.e., the candidate gene name);
- the candidate gene identifier;
- the preceding and the following two words;
- the number of words in the matched phrase;
- the total number of possible gene identifiers for the matched phrase;
- all character prefixes and suffixes of length up to four for the words within the phrase;
- the words from the current sentence;
- the lemmata of the words from the current sentence;
- presence of the matched phrase in various lexicons;
- presence of sequences of words from the current sentence in various lexicons.

These predicates are used in features like this:

$$f_i(x,y) = \begin{cases} \mathbf{1} & \textbf{if } 'WORD_{-1} = confers' \in x, \\ & \mathbf{y} = \text{human-positive}; \\ \mathbf{0} & \textbf{otherwise}. \end{cases}$$

Some of the above features have been used in previous research in normalizing gene mentions [5, 21], but some are novel, e.g., words/lemmata from the current sentence and presence of sequences of words from the current sentence in various lexicons.

The lexicons we used were compiled from the UniprotKB part of Uniprot[6], Entrez Gene[7], Entrez Taxonomy[8], and various disease databases. From UniprotKB and Entrez Gene, we took the gene or gene product names, and we filtered them based on the organism: human, mouse, yeast or fly. From Entrez Organism, we compiled an organism list. From the disease databases, we compiled a list of diseases in human and mouse. The different lexicons had different matching scopes. For example, while the lexicon of gene names was used to match both against the candidate gene mention and against the rest of the sentence, organism and diseases lists were only allowed to match against the rest of the sentence.

[6] http://www.uniprot.org/
[7] http://www.ncbi.nlm.nih.gov/sites/entrez?db=gene
[8] http://www.ncbi.nlm.nih.gov/Taxonomy/

Next, we created features from the above predicates. In the process of doing so, we allowed for some combinations of simple labels and predicates to co-appear in the feature vectors. In particular, this allowed for features to co-appear in human and mouse instances if they shared predicates: the main intuition is that positive examples for mouse and human should naturally share some positive features and should only differ in organism-specific features. We have achieved this by means of feature function decomposition as will be described below.

First, note that all machine learning algorithms used in the present work are linear models. This means that for each input example x, the best output label $\hat{y}$ can be found using an inference procedure that can be expressed by the following equation:

$$\hat{y} = \arg\max_y [w^T \cdot f(x,y)] = \arg\max_y \sum_i^m w_i f_i(x,y) \quad (2)$$

where $f(x,y)$ is a vector of feature functions, w is a weight vector, $f_i(x,y)$ and x_i are the values of the i^{th} coordinates of those vectors, and y ranges over the possible labels.

In our joint model, y can take the following three possible values: *human-positive*, *mouse-positive*, and *negative*. We further decomposed *human-positive* and *mouse-positive* into three simplified labels: *human*, *mouse* and *positive*. In this new representation, the simplified *positive* label co-exists in both *human-positive* and *mouse-positive*. Therefore, it will be useful if some of the features between the two instances – one for mouse and one for human – are shared since they both represent positive gene matches. In order to achieve this, we decomposed the feature function $f_i(x, \textit{human-positive})$ as follows:

$$f_i(x, \textit{human-positive}) = f_{i,1}(f_{i,2}(x, \textit{positive}), \textit{human-positive}) \quad (3)$$

where $f_{i,1}$ maps the input into a sparse vector, and $f_{i,2}$ combines it with a possible output in order to generate the final sparse vector used to assess the compatibility of the label for this input. In this representation, all instances labeled *human-positive* and *mouse-positive* share some features since they use a common feature sub-function $f_{i,2}(x, \textit{positive})$.

Note that we only allowed a subset of the predicates listed above to participate in the feature sub-function – those that are organism-independent, e.g., the number of words in the matched phrase, the total number of possible gene identifiers for the matched phrase, all character prefixes and suffixes of length up to four for the words within the phrase, etc. For example, the following feature for $f_{i,2}$ (see Eq 3 above) can be generated both for a *human-positive* and for a *mouse-positive* instance:

$$f_{i,2}(x,y) = \begin{cases} \mathbf{1} & \textbf{if } '\#(\text{GenesMatched}) = 5' \in x, \\ & \mathbf{y} = \text{positive}; \\ \mathbf{0} & \textbf{otherwise}. \end{cases}$$

4 Experiments and evaluation

Below we describe our experiments in evaluating the joint model described in Section 3.4 on the standard test sets of BioCreAtIvE. We tried Naïve Bayes, Maximum Entropy, Perceptron, MIRA, as well as averaged versions of the last two. Since the training data for this task were limited, we also studied the dependence of each classifier on the number of training examples from both manually annotated and noisy data sources.

Figure 1 shows the performance of our six classifiers as a function of the number of manually annotated training examples for mouse. Maximum Entropy, averaged MIRA and Perceptron outperformed the rest by 3-4% of F_1, in the range of 2,000-2,600 training examples. For the case of limited training data, in the range of 100-200 examples, the best-scoring classifier was Maximum Entropy.

As Figure 2 shows, for the human training data, in the range of 2,000-3,000 manually annotated examples, the best classifiers were again Maximum Entropy and the averaged Perceptron, and for 100-200 training examples, Maximum Entropy and averaged MIRA were tied for the first place. The Naïve Bayes classifier was the worse-performing one for both the 100-200 and 2,000-3,000 ranges.

As a second set of experiments, we combined the mouse and the human training examples, and we used a multi-class version of the learning schemata to train and evaluate the joint mouse-human statistical model that has been described above. Figure 3 shows the performance for different numbers of training examples for the joint model. Again, the Maximum Entropy and the averaged Perceptron outperformed the remaining classifiers in the full range of numbers of training examples; Perceptron scored third in this experiment.

Table 2 shows the data for Maximum Entropy, Naïve Bayes, Perceptron and MIRA presented already in detail on Figure 3, e.g., the evaluation is presented separately for mouse and human and in terms of precision, recall and F_1-measure. Note that all learning methods show well-balanced precision and recall, which is a very desirable property for a gene mention normalization system. The best performing classifier for the joint model when tested on human examples was Maximum Entropy – it outperformed the rest by more than 2% absolute difference in F_1-measure. For mouse, MIRA was the best, directly followed by Maximum Entropy.

In order to boost the performance of the classifier even further, we added to the model the additional noisy training examples that were provided by the BioCreAtIvE organizers for both human and mouse. For this set of experiments, we selected the Maximum Entropy classifier, and we achieved an absolute increase in F_1 score of more than 12% and 7% for mouse and human test examples respectively (see Table 3, column *A*).

In our last experiment, we used the feature function decomposition as described in Section 3.4, which resulted in further improvement to reach the final F_1 score for the Maximum Entropy classifier of 97.09% and 97.64% for mouse and human respectively (see Table 3, Column *B*).

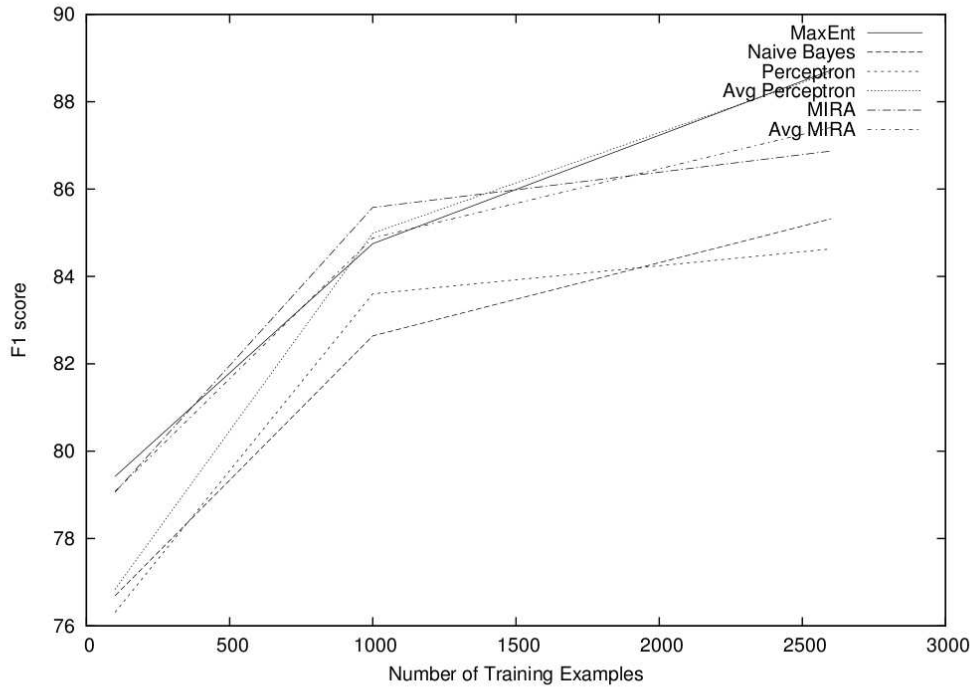

Fig. 1: *Normalization of mouse gene mentions using different classifiers.*

	Maximum Entropy			Naïve Bayes			Perceptron			MIRA		
#examples	R	P	F_1	R	P	F_1	R	P	F_1	R	P	F_1
mouse												
2400	81.25	81.95	81.60	87.66	70.45	78.12	80.04	76.97	78.48	84.15	74.43	79.00
5600	80.80	87.82	84.16	87.20	77.85	82.26	84.80	81.53	83.13	85.60	86.29	85.94
human												
2400	81.84	82.51	82.17	87.26	79.18	83.02	80.39	77.93	79.14	74.16	82.47	78.10
5600	90.57	89.28	89.92	92.75	83.11	87.67	84.05	89.92	86.89	93.47	74.56	82.95

Table 2: *Evaluation of the joint human-mouse model with different classifiers. Precision (P), recall (R) and F_1-measure are shown in %.*

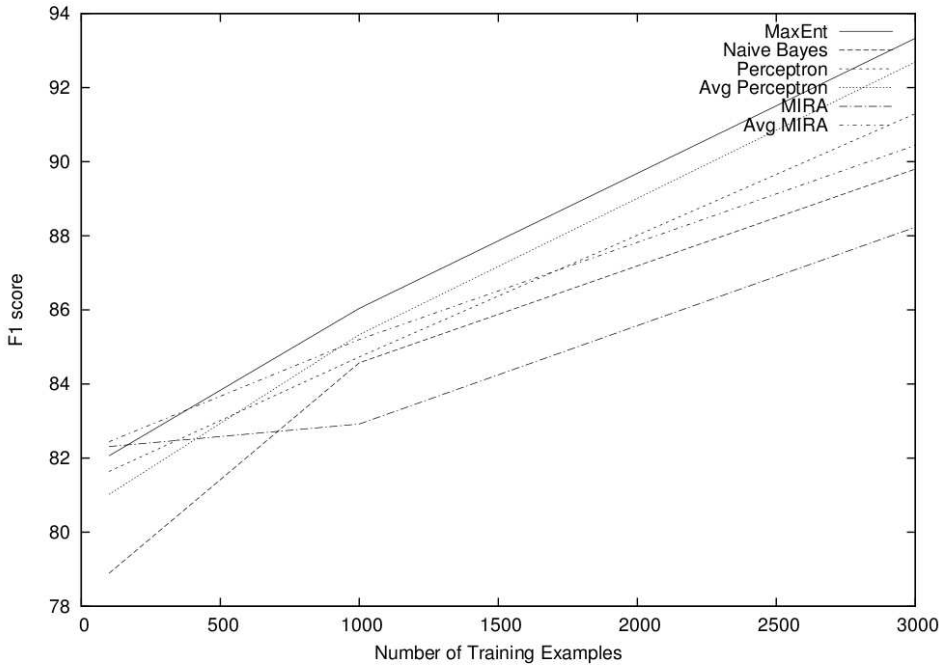

Fig. 2: *Normalization of human gene mentions using different classifiers.*

	mouse			human		
	R	P	F_1	R	P	F_1
A	96.64	96.11	96.37	96.84	97.42	97.13
B	96.62	97.61	97.09	96.04	98.03	97.64

Table 3: *Evaluation of the Maximum Entropy joint human-mouse model, with regular (*A*) and decomposed (*B*) feature functions. Precision (P), recall (R) and F_1-measure are shown in %.*

5 Conclusions and future work

We have proposed an extension to the popular task of gene mention normalization to also recognize whether the target gene mention is valid and to find the organism it is from. We have addressed this extended task using a joint model for gene and organism name normalization which allows for instances from different organisms to share features. This model yielded sizable performance gains with the following six statistical classifiers: Naïve Bayes, Maximum Entropy, Perceptron and MIRA, as well as averaged versions of the last two. The evaluation results for our best joint classifier using Maximum Entropy and noisy training data show F_1 score of over 97%, which proves the potential of the approach.

Unlike previous work, we have focused on training examples preparation and classification – two stages that are often underestimated when designing gene name normalization systems. We have shown that by tuning the loss function of the structured MIRA classifier, it is possible to enhance the recall of the gene tagger significantly. We have further proposed and carefully evaluated a joint model that recognizes both the organism and the correctness of a gene mention - gene identifier pair in a particular text context. Finally, we have evaluated six classifiers, comparing them on training data of different sizes, and we have shown that the performance of several of them is very close when the number of training examples is in the range 2,000-3,000.

There are many ways in which the present work can be extended in the future. First, we would like to experiment with more training data, including noisy and unlabeled data, in order to reveal the full potential of the idea for feature function decomposition. Applying the idea to other related problems in the biomedical domain is another promising research direction. Ultimately, we will try to integrate the joint model into a fully functional system and compare its performance to that of existing gene mention normalization systems for a single organism, e.g., those that participated in BioCreAtIvE. We further plan to include other important organisms in the joint model, the most obvious candidates being yeast and fly.

Acknowledgments

The work reported in this paper was partially supported by the EU FP7 project 215535 LarKC.

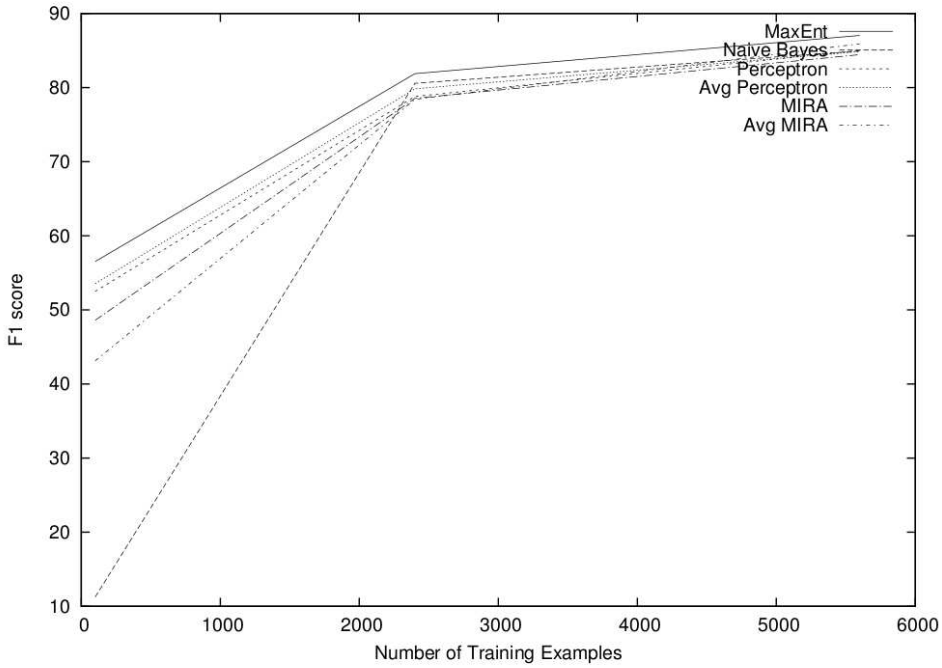

Fig. 3: *Joint organism and gene normalization using different classifiers.*

References

[1] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, 1996.

[2] K. B. Cohen, G. K. Acquaah-Mensah, A. E. Dolbey, and L. Hunter. Contrast and variability in gene names. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain*, pages 14–20, 2002.

[3] K. Crammer. *Online Learning of Complex Categorial Problems.* PhD thesis, Hebrew Univeristy of Jerusalem, 2004.

[4] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585, 2006.

[5] J. Crim, R. McDonald, and F. Pereira. Automatically annotating documents with normalized gene lists. *BMC Bioinformatics*, 6 Suppl 1:S13, 2005.

[6] Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. In *Machine Learning*, pages 277–296, 1999.

[7] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck. Prominer: rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6 Suppl 1:S14, 2005.

[8] L. Hirschman, M. Colosimo, A. Morgan, and A. Yeh. Overview of BioCreAtIvE task 1B: normalized gene lists. *BMC Bioinformatics*, 6 Suppl 1:S11, 2005.

[9] M. A. Jaro. Probabilistic linkage of large public health data file. *Statistics in Medicine*, 14:491–498, 1995.

[10] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*. Morgan Kaufmann, 2001.

[11] M. E. Maron. Automatic indexing: An experimental inquiry. *J. ACM*, 8(3):404–417, 1961.

[12] A. McCallum. Efficiently inducing features of conditional random fields. In *Proceedings of UAI*, 2003.

[13] R. McDonald, K. Crammer, and F. Pereira. Online large-margin training of dependency parsers. In *Proceedings of ACL*. ACL, 2005.

[14] R. McDonald and F. Pereira. Identifying gene and protein mentions in text using conditional random fields. *BMC Bioinformatics*, (Suppl 1):S6(6), 2005.

[15] A. A. Morgan, L. Hirschman, M. Colosimo, A. S. Yeh, and J. B. Colombe. Gene name identification and normalization using a model organism database. *J Biomed Inform*, 37(6):396–410, Dec 2004.

[16] A. A. Morgan, Z. Lu, X. Wang, A. M. Cohen, J. Fluck, P. Ruch, A. Divoli, K. Fundel, R. Leaman, J. Hakenberg, C. Sun, H. hui Liu, R. Torres, M. Krauthammer, W. W. Lau, H. Liu, C.-N. Hsu, M. Schuemie, K. B. Cohen, and L. Hirschman. Overview of BioCreative II gene normalization. *Genome Biol*, 9 Suppl 2:S3, 2008.

[17] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 1989.

[18] L. Ramshaw and M. Marcus. Text chunking using transformation-based learning. In D. Yarovsky and K. Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, Somerset, New Jersey, 1995. ACL.

[19] H. ren Fang, K. Murphy, Y. Jin, J. S. Kim, and P. S. White. Human gene name normalization using text matching with automatically extracted synonym dictionaries. In *BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL*, pages 41–48, 2006.

[20] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.

[21] B. Wellner. Weakly supervised learning methods for improving the quality of gene name normalization data. In *ACL-ISMB workshop on linking literature, information and knowledge for biology*, 2005.