ACL-08: HLT

BioNLP 2008

Current Trends
in
Biomedical Natural Language Processing

Proceedings of the Workshop

June 19, 2008
The Ohio State University
Columbus, Ohio, USA

Production and Manufacturing by
Omnipress Inc.
2600 Anderson Street
Madison, WI 53707
USA

Sponsored by:

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-932432-11-4

Current trends in biomedical natural language processing: the view from computational linguistics

Dina Demner-Fushman, Sophia Ananiadou, K. Bretonnel Cohen, John Pestian, Jun'ichi Tsujii, and Bonnie Webber

Background

Research in computational linguistics in the biomedical domain traditionally focuses on two major areas: fundamental advances in language processing; and application of language processing methods to bridge the gap between basic biomedical research, clinical research, and translation of both types of research into practice. Several conferences provide opportunities for discussion of these two types of research in specific sub-domains of Biomedical Natural Language Processing. For example, Intelligent Systems for Molecular Biology (ISMB) and its associated special interest group and Pacific Symposium on Biocomputing (PSB) focus on NLP research applied to issues of interest to biologists, whereas American Medical Informatics Association (AMIA) is concerned with medical informatics issues.

Rather than focusing on a specific area of interest, ACL BioNLP workshop strives to provide a forum for any important, new, and exciting research in the field of Biomedical Natural Language Processing. Rather than focusing on a specific theme as we have in previous years, the goal of the workshop this year was to solicit work of interest to NLP researchers on any topic in the biomedical domain.

Submissions, acceptance, and themes

Asking researchers to share their interests was rewarded by 34 submissions (5 posters and 19 full papers). Of those, 10 were accepted as full papers and 18 as poster presentations. The combined expertise of the program committee allowed for providing three thorough reviews for each paper. The exceptionally high quality manuscripts accepted for presentation cover a wide area of subjects in clinical and biological areas, as well as methodological issues applicable to both sublanguages.

Named entity recognition (NER) continues to be an active area of research. NER research presented here involves development of new statistical and hybrid approaches to identification and disambiguation of gene [1], protein [2], chemical names [3], and clinical entities.

Overwhelmingly, researchers chose statistical or hybrid approaches to the tasks at hand. This is probably the reason for growing interest in creation of annotated corpora [4], development of methods for augmenting the existing annotation [5], speeding up the annotation process [5], and reducing its cost; evaluating the comparability of results obtained applying the same methods to different collections [6], And increasing compatibility of different annotations [7].

Increasingly sophisticated relation extraction methods [6, 8] are being applied to a broader set of

relations [9]. Other steps towards deeper understanding of the text include methods for creation of gene profiles [10], identification of uncertainty [11], discourse connectivity [12], and temporal features of clinical conditions [13].

The applicability of NLP methods to clinical tasks is explored in the work on identification of language impairments [14] and seriousness of suicidal attempts [15].

Finally, application of NLP methods to classic information retrieval problems such as automatic indexing of biomedical literature [16] and the newer information retrieval problem of image retrieval [17] are explored.

Acknowledgments

Organizing the BioNLP workshop is an extremely gratifying experience. We are indebted to the authors who chose to submit their high quality research covering a variety of interesting topics to this workshop. Our main hurdle was the selection of oral presentations, in which we relied on the thoughtful and thorough reviews provided by the program committee. We thank the ACL organizers for their help and clarifications on numerous issues. Last, but not least our thanks go to the workshop sponsors: The Computational Medicine Center and Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center and The UK National Centre for Text Mining (NaCTeM).

References

[1] Xinglong Wang and Michael Mattews. *Species Disambiguation for Biomedical Term Identification. BioNLP* 2008.

[2] Yutaka Sasaki, Yoshimasa Tsuruoka, John McNaught and Sophia Ananiadou. *How to Make the Most of NE Dictionaries in Statistical NER. BioNLP* 2008.

[3] Peter Corbett and Ann Copestake. *Cascaded Classifiers for Confidence-Based Chemical Named Entity Recognition. BioNLP* 2008.

[4] György Szarvas, Veronika Vincze, Richárd Farkas and János Csirik. *The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts. BioNLP* 2008.

[5] Yoshimasa Tsuruoka, Jun'ichi Tsujii and Sophia Ananiadou. *Accelerating the Annotation of Sparse Named Entities by Dynamic Sentence Selection. BioNLP* 2008.

[6] Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter and Tapio Salakoski. *A Graph Kernel for Protein-Protein Interaction Extraction. BioNLP* 2008.

[7] Yue Wang, Kazuhiro Yoshida, Jin-Dong Kim, Rune Saetre and Jun'ichi Tsujii. *Raising the Compatibility of Heterogeneous Annotations: A Case Study on Protein Mention Recognition. BioNLP* 2008.

[8] Barry Haddow. *Using Automated Feature Optimisation to Create an Adaptable Relation Extraction System. BioNLP* 2008.

[9] Angus Roberts, Robert Gaizauskas and Mark Hepple. *Extracting Clinical Relationships from Patient Narratives. BioNLP* 2008.

[10] Catalina O Tudor, K Vijay-Shanker and Carl J Schmidt. *Mining the Biomedical Literature for Genic Information. BioNLP* 2008.

[11] Halil Kilicoglu and Sabine Bergler. *Recognizing Speculative Language in Biomedical Research Articles: A Linguistically Motivated Perspective. BioNLP* 2008.

[12] Hong Yu, Nadya Frid, Susan McRoy, Rashmi Prasad, Alan Lee and Aravind Joshi. *A Pilot Annotation to Investigate Discourse Connectivity in Biomedical Text. BioNLP* 2008.

[13] Danielle Mowery, Henk Harkema and Wendy Chapman. *Temporal Annotation of Clinical Text. BioNLP* 2008.

[14] Thamar Solorio and Yang Liu. *Using Language Models to Identify Language Impairment in Spanish-English Bilingual Children. BioNLP* 2008.

[15] John Pestian, Pawel Matykiewicz, Jacqueline Grupp-Phelan, Sarah Arszman Lavanier, Jennifer Combs, and Robert Kowatch. *Distinguishing between completer and ideator suicide notes: A comparison of machine learning methods. BioNLP* 2008.

[16] Aurelie Neveol, Sonya Shooshan and Vincent Claveau. *Automatic inference of indexing rules for MEDLINE. BioNLP* 2008.

[17] Paul Buitelaar, Pinar Oezden Wennerberg and Sonja Zillner. *Statistical Term Profiling for Query Pattern Mining. BioNLP* 2008.

Organizers:

Dina Demner-Fushman, US National Library of Medicine
Sophia Ananiadou, National Centre for Text Mining and University of Manchester, UK
Kevin Bretonnel Cohen, The MITRE Corporation and University of Colorado School of Medicine
John Pestian, Computational Medicine Center, Cincinnati Childrens Hospital and Medical Center
Jun'ichi Tsujii, University of Tokyo, Japan and University of Manchester, UK
Bonnie Webber, University of Edinburgh, UK

Program Committee:

Alan Aronson, LHNCBC, US National Library of Medicine
Catherine Blake, University of North Carolina
Olivier Bodenreider, LHNCBC, US National Library of Medicine
Bob Carpenter, Alias-i
Wendy Chapman, University of Pittsburgh
Aaron Cohen, Oregon Health and Science University
Nigel Collier, National Institute of Informatics, Tokyo

Noemie Elhadad, Columbia University
Marcelo Fiszman, LHNCBC, US National Library of Medicine
Kristofer Franzén, Swedish Institute of Computer Science
Carol Friedman, Columbia College of Physicians and Surgeons
Peter Haug, University of Utah
Marti Hearst, University of California at Berkeley
Su Jian, A-star
Jin-Dong Kim, University of Tokyo
Marc Light, Thomson
Zhiyong Lu, NCBI, US National Library of Medicine
Aurelie Neveol, LHNCBC, US National Library of Medicine
Serguei Pakhomov, University of Minnesota
Thomas Rindflesch, LHNCBC, US National Library of Medicine
Daniel Rubin, Stanford University
Hagit Shatkay, Queen's University, Canada
Larry Smith, NCBI, US National Library of Medicine
Yuka Tateisi, University of Tokyo
Yoshimasa Tsuruoka, University of Manchester
Alfonso Valencia, Centro Nacional de Biotecnologia
Karin Verspoor, Center for Computational Pharmacology, University of Colorado School of Medicine
Peter White, Children's Hospital of Philadelphia
W. John Wilbur, NCBI, US National Library of Medicine
Limsoon Wong, National University of Singapore
Hong Yu, University of Wisconsin
Pierre Zweigenbaum, LIMSI

Invited Speakers:

John J. Hutton, MD Senior Vice President, Biomedical Informatics,
Cincinnati Children's Hospital Medical Center,
University of Cincinnati College of Medicine

Hon S. Pak, MD Chief, Advanced Information Technology Group,
Telemedicine & Advanced Technology Research Center (TATRC)

Table of Contents

A Graph Kernel for Protein-Protein Interaction Extraction
Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter and Tapio Salakoski . . . 1

Extracting Clinical Relationships from Patient Narratives
Angus Roberts, Robert Gaizauskas and Mark Hepple 10

Using Automated Feature Optimisation to Create an Adaptable Relation Extraction System
Barry Haddow 19

Mining the Biomedical Literature for Genic Information
Catalina O Tudor, K Vijay-Shanker and Carl J Schmidt 28

Accelerating the Annotation of Sparse Named Entities by Dynamic Sentence Selection
Yoshimasa Tsuruoka, Jun'ichi Tsujii and Sophia Ananiadou 30

The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts
György Szarvas, Veronika Vincze, Richárd Farkas and János Csirik 38

Recognizing Speculative Language in Biomedical Research Articles: A Linguistically Motivated Perspective
Halil Kilicoglu and Sabine Bergler 46

Cascaded Classifiers for Confidence-Based Chemical Named Entity Recognition
Peter Corbett and Ann Copestake 54

How to Make the Most of NE Dictionaries in Statistical NER
Yutaka Sasaki, Yoshimasa Tsuruoka, John McNaught and Sophia Ananiadou 63

Species Disambiguation for Biomedical Term Identification
Xinglong Wang and Michael Mattews 71

Knowledge Sources for Word Sense Disambiguation of Biomedical Text
Mark Stevenson, Yinkun Guo, Robert Gaizauskas and David Martinez 80

Automatic inference of indexing rules for MEDLINE
Aurelie Neveol, Sonya Shooshan and Vincent Claveau 88

Prediction of Protein Sub-cellular Localization using Information from Texts and Sequences.
Hong-Woo Chun, Chisato Yamasaki, Naomi Saichi, Masayuki Tanaka, Teruyoshi Hishiki, Tadashi Imanishi, Takashi Gojobori, Jin-Dong Kim, Jun'ichi Tsujii and Toshihisa Takagi 90

A Pilot Annotation to Investigate Discourse Connectivity in Biomedical Text
Hong Yu, Nadya Frid, Susan McRoy, Rashmi Prasad, Alan Lee and Aravind Joshi 92

Conditional Random Fields and Support Vector Machines for Disorder Named Entity Recognition in Clinical Texts
Dingcheng Li, Guergana Savova and Karin Kipper-Schuler 94

Using Natural Language Processing to Classify Suicide Notes
John Pestian, Pawel Matykiewicz, Jacqueline Grupp-Phelan, Sarah Arszman Lavanier, Jennifer Combs and Robert Kowatch 96

Extracting Protein-Protein Interaction based on Discriminative Training of the Hidden Vector State Model
Deyu Zhou and Yulan He 98

A preliminary approach to extract drugs by combining UMLS resources and USAN naming conventions
Isabel Segura-Bedmar, Paloma Martínez and Doaa Samy 100

Mapping Clinical Notes to Medical Terminologies at Point of Care
Yefeng Wang and Jon Patrick 102

An Approach to Reducing Annotation Costs for BioNLP
Michael Bloodgood and K Vijay-Shanker 104

Temporal Annotation of Clinical Text
Danielle Mowery, Henk Harkema and Wendy Chapman 106

CBR-Tagger: a case-based reasoning approach to the gene/protein mention problem
Mariana Neves, Monica Chagoyen, José María Carazo and Alberto Pascual-Montano 108

Textual Information for Predicting Functional Properties of the Genes
Oana Frunza and Diana Inkpen 110

Determining causal and non-causal relationships in biomedical text by classifying verbs using a Naive Bayesian Classifier
Pieter van der Horn, Bart Bakker, Gijs Geleijnse, Jan Korst and Sergei Kurkin 112

Statistical Term Profiling for Query Pattern Mining
Paul Buitelaar, Pinar Oezden Wennerberg and Sonja Zillner 114

Using Language Models to Identify Language Impairment in Spanish-English Bilingual Children
Thamar Solorio and Yang Liu 116

Raising the Compatibility of Heterogeneous Annotations: A Case Study on
Yue Wang, Kazuhiro Yoshida, Jin-Dong Kim, Rune Saetre and Jun'ichi Tsujii 118

Adaptive Information Extraction for Complex Biomedical Tasks
Donghui Feng, Gully Burns and Eduard Hovy 120

Workshop Program

Thursday, June 19, 2008

8:45–8:50 Opening Remarks

Session 1: Relations Extraction and Text Mining

8:50–9:15 *A Graph Kernel for Protein-Protein Interaction Extraction*
Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter and Tapio Salakoski

9:15–9:40 *Extracting Clinical Relationships from Patient Narratives*
Angus Roberts, Robert Gaizauskas and Mark Hepple

9:40–10:05 *Using Automated Feature Optimisation to Create an Adaptable Relation Extraction System*
Barry Haddow

10:05–10:30 *Mining the Biomedical Literature for Genic Information*
Catalina O Tudor, K Vijay-Shanker and Carl J Schmidt

10:30–11:00 Coffee break

11:00–11:40 Invited Talk by John Hutton, MD

Session 2: Annotation Issues and Uncertainty Detection

11:45–12:10 *Accelerating the Annotation of Sparse Named Entities by Dynamic Sentence Selection*
Yoshimasa Tsuruoka, Jun'ichi Tsujii and Sophia Ananiadou

12:10–12:35 *The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts*
György Szarvas, Veronika Vincze, Richárd Farkas and János Csirik

12:35–13:00 *Recognizing Speculative Language in Biomedical Research Articles: A Linguistically Motivated Perspective*
Halil Kilicoglu and Sabine Bergler

13:00–14:00 Lunch

Thursday, June 19, 2008 (continued)

Session 3: Named Entity Recognition

14:00–14:25 *Cascaded Classifiers for Confidence-Based Chemical Named Entity Recognition*
Peter Corbett and Ann Copestake

14:25–14:50 *How to Make the Most of NE Dictionaries in Statistical NER*
Yutaka Sasaki, Yoshimasa Tsuruoka, John McNaught and Sophia Ananiadou

14:50–15:30 Invited Talk by Hon Pak, MD

15:30–16:00 Coffee break

Session 4: Word Sense Disambiguation

16:00–16:25 *Species Disambiguation for Biomedical Term Identification*
Xinglong Wang and Michael Mattews

16:25–16:50 *Knowledge Sources for Word Sense Disambiguation of Biomedical Text*
Mark Stevenson, Yinkun Guo, Robert Gaizauskas and David Martinez

17:00–18:00 Poster Session

Automatic inference of indexing rules for MEDLINE
Aurelie Neveol, Sonya Shooshan and Vincent Claveau

Prediction of Protein Sub-cellular Localization using Information from Texts and Sequences.
Hong-Woo Chun, Chisato Yamasaki, Naomi Saichi, Masayuki Tanaka, Teruyoshi Hishiki, Tadashi Imanishi, Takashi Gojobori, Jin-Dong Kim, Jun'ichi Tsujii and Toshihisa Takagi

A Pilot Annotation to Investigate Discourse Connectivity in Biomedical Text
Hong Yu, Nadya Frid, Susan McRoy, Rashmi Prasad, Alan Lee and Aravind Joshi

Conditional Random Fields and Support Vector Machines for Disorder Named Entity Recognition in Clinical Texts
Dingcheng Li, Guergana Savova and Karin Kipper-Schuler

Using Natural Language Processing to Classify Suicide Notes
John Pestian, Pawel Matykiewicz, Jacqueline Grupp-Phelan, Sarah Arszman Lavanier, Jennifer Combs and Robert Kowatch

Thursday, June 19, 2008 (continued)

Extracting Protein-Protein Interaction based on Discriminative Training of the Hidden Vector State Model
Deyu Zhou and Yulan He

A preliminary approach to extract drugs by combining UMLS resources and USAN naming conventions
Isabel Segura-Bedmar, Paloma Martínez and Doaa Samy

Mapping Clinical Notes to Medical Terminologies at Point of Care
Yefeng Wang and Jon Patrick

An Approach to Reducing Annotation Costs for BioNLP
Michael Bloodgood and K Vijay-Shanker

Temporal Annotation of Clinical Text
Danielle Mowery, Henk Harkema and Wendy Chapman

CBR-Tagger: a case-based reasoning approach to the gene/protein mention problem
Mariana Neves, Monica Chagoyen, José María Carazo and Alberto Pascual-Montano

Textual Information for Predicting Functional Properties of the Genes
Oana Frunza and Diana Inkpen

Determining causal and non-causal relationships in biomedical text by classifying verbs using a Naive Bayesian Classifier
Pieter van der Horn, Bart Bakker, Gijs Geleijnse, Jan Korst and Sergei Kurkin

Statistical Term Profiling for Query Pattern Mining
Paul Buitelaar, Pinar Oezden Wennerberg and Sonja Zillner

Using Language Models to Identify Language Impairment in Spanish-English Bilingual Children
Thamar Solorio and Yang Liu

Raising the Compatibility of Heterogeneous Annotations: A Case Study on
Yue Wang, Kazuhiro Yoshida, Jin-Dong Kim, Rune Saetre and Jun'ichi Tsujii

Adaptive Information Extraction for Complex Biomedical Tasks
Donghui Feng, Gully Burns and Eduard Hovy

A Graph Kernel for Protein-Protein Interaction Extraction

Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter and Tapio Salakoski
Turku Centre for Computer Science
and Department of IT, University of Turku
Joukahaisenkatu 3-5
20520 Turku, Finland
firstname.lastname@utu.fi

Abstract

In this paper, we propose a graph kernel based approach for the automated extraction of protein-protein interactions (PPI) from scientific literature. In contrast to earlier approaches to PPI extraction, the introduced all-dependency-paths kernel has the capability to consider full, general dependency graphs. We evaluate the proposed method across five publicly available PPI corpora providing the most comprehensive evaluation done for a machine learning based PPI-extraction system. Our method is shown to achieve state-of-the-art performance with respect to comparable evaluations, achieving 56.4 F-score and 84.8 AUC on the AImed corpus. Further, we identify several pitfalls that can make evaluations of PPI-extraction systems incomparable, or even invalid. These include incorrect cross-validation strategies and problems related to comparing F-score results achieved on different evaluation resources.

1 Introduction

Automated protein-protein interaction (PPI) extraction from scientific literature is a task of significant interest in the BioNLP field. The most commonly addressed problem has been the extraction of binary interactions, where the system identifies which protein pairs in a sentence have a biologically relevant relationship between them. Proposed solutions include both hand-crafted rule-based systems and machine learning approaches (see e.g. (Bunescu et al., 2005)). A wide range of results have been reported for the systems, but as we will show, differences in evaluation resources, metrics and strategies make direct comparison of these numbers problematic. Further, the results gained from the BioCreative II evaluation, where the best performing system achieved a 29% F-score (Hunter et al., 2008), suggest that the problem of extracting binary protein protein interactions is far from solved.

The public availability of large annotated PPI-corpora such as AImed (Bunescu et al., 2005), BioInfer (Pyysalo et al., 2007a) and GENIA (Kim et al., 2008), provides an opportunity for building PPI extraction systems automatically using machine learning. A major challenge is how to supply the learner with the contextual and syntactic information needed to distinguish between interactions and non-interactions. To address the ambiguity and variability of the natural language expressions used to state PPI, several recent studies have focused on the development, adaptation and application of NLP tools for the biomedical domain. Many high-quality domain-specific tools are now freely available, including full parsers such as that introduced by Charniak and Lease (2005). Additionally, a number of conversions from phrase structure parses to dependency structures that make the relationships between words more directly accessible have been introduced. These include conversions into representations such as the Stanford dependency scheme (de Marneffe et al., 2006) that are explicitly designed for information extraction purposes. However, specialized feature representations and kernels are required to make learning from such structures possible.

Approaches such as subsequence kernels (Bunescu and Mooney, 2006), tree kernels (Zelenko

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 1–9,
Columbus, Ohio, USA, June 2008.

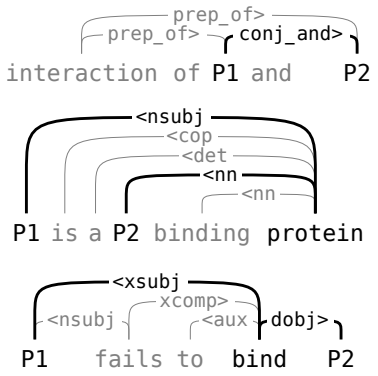

Figure 1: Stanford dependency parses ("collapsed" representation) where the shortest path, shown in bold, excludes important words.

et al., 2003) and shortest path kernels (Bunescu and Mooney, 2005) have been proposed and successfully used for relation extraction. However, these methods lack the expressive power to consider representations derived from general, possibly cyclic, dependency graph structures, such as those generated by the Stanford tools. The subsequence kernel approach does not consider parses at all, and the shortest path approach is limited to representing only a single path in the full dependency graph, which excludes relevant words even in many simple cases (Figure 1). Tree kernels can represent more complex structures, but are still restricted to tree representations.

Lately, in the framework of kernel-based machine learning methods there has been an increased interest in designing kernel functions for graph data. Building on the work of Gärtner et al. (2003), graph representations tailored for the task of dependency parse ranking were proposed by Pahikkala et al. (2006b). Though the proposed representations are not directly applicable to the task of PPI extraction, they offer insight in how to learn from dependency graphs. We develop a graph kernel approach for PPI extraction based on these ideas.

We next define a graph representation suitable for describing potential interactions and introduce a kernel which makes efficient learning from a general, unrestricted graph representation possible. Then we provide a short description of the sparse regularized least squares (sparse RLS) kernel-based machine learning method we use for PPI-extraction. Further, we rigorously assess our method on five publicly available PPI corpora, providing the first broad cross-corpus evaluation with a machine learning approach to PPI extraction. Finally, we discuss the effects that different evaluation strategies, choice of corpus and applied metrics have on measured performance, and conclude.

2 Method

We next present our graph representation, formalize the notion of graph kernels, and present our learning method of choice, the sparse RLS.

2.1 Graph encoding of sentence structure

As in most recent work on machine learning for PPI extraction, we cast the task as learning a decision function that determines for each unordered candidate pair of protein names occurring together in a sentence whether the two proteins interact. In the following, we first define the graph representation used to represent an interaction candidate pair. We then proceed to derive the kernel used to measure the similarities of these graphs.

We assume that the input of our learning method is a dependency parse of a sentence where a pair of protein names is marked as the candidate interaction for which an extraction decision must be made. Based on this, we form a weighted, directed graph that consists of two unconnected subgraphs. One represents the dependency structure of the sentence, and the other the linear order of the words (see Figure 2).

The first subgraph is built from the dependency analysis. One vertex and an associated set of labels is created in the graph for each token and for each dependency. The vertices that represent tokens have as labels the text and part-of-speech (POS) of the token. To ensure generalization of the learned extraction model, the labels of vertices that correspond to protein names are replaced with PROT1, PROT2 or PROT, where PROT1 and PROT2 are the pair of interest. The vertices that represent dependencies are labeled with the type of the dependency. The edges in the subgraph are defined so that each dependency vertex is connected by an incoming edge from the vertex representing its governor token, and by an outgoing edge to the vertex representing its de-

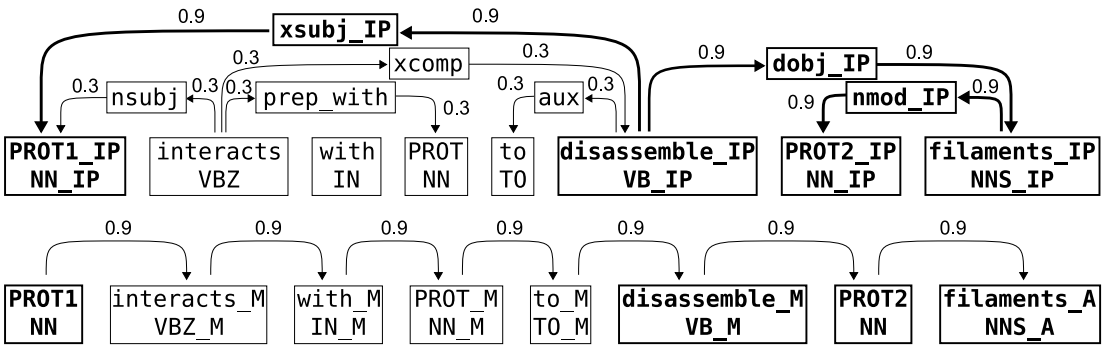

Figure 2: Graph representation generated from an example sentence. The candidate interaction pair is marked as PROT1 and PROT2, the third protein is marked as PROT. The shortest path between the proteins is shown in bold. In the dependency based subgraph all nodes in a shortest path are specialized using a post-tag (IP). In the linear order subgraph possible tags are (B)efore, (M)iddle, and (A)fter. For the other two candidate pairs in the sentence, graphs with the same structure but different weights and labels would be generated.

pendent token. The graph thus represents the entire sentence structure.

It is widely acknowledged that the words between the candidate entities or connecting them in a syntactic representation are particularly likely to carry information regarding their relationship; (Bunescu and Mooney, 2005) formalize this intuition for dependency graphs as the *shortest path hypothesis*. We apply this insight in two ways in the graph representation: the labels of the nodes on the shortest undirected paths connecting PROT1 and PROT2 are differentiated from the labels outside the paths using a special tag. Further, the edges are assigned weights; after limited preliminary experiments, we chose a simple weighting scheme where all edges on the shortest paths receive a weight of 0.9 and other edges receive a weight of 0.3. The representation thus allows us to emphasize the shortest path without completely disregarding potentially relevant words outside of the path.

The second subgraph is built from the linear structure of the sentence. For each token, a second vertex is created and the labels for the vertices are derived from the texts, POS-tags and named entity tagging as above. The labels of each word are specialized to denote whether the word appears before, in-between, or after the protein pair of interest. Each word node is connected by an edge to its succeeding word, as determined by sentence order the of the words. Each edge is given the weight 0.9.

2.2 The all-dependency-paths graph kernel

We next formalize the graph representation and present the all-dependency-paths kernel. This kernel can be considered as a practical instantiation of the theoretical graph kernel framework introduced by Gärtner et al. (2003). Let V be the set of vertices in the graph and $\mathcal{L}$ be the set of possible labels vertices can have. We represent the graph with an adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$, whose rows and columns are indexed by the vertices, and $[A]_{i,j}$ contains the weight of the edge connecting $v_i \in V$ and $v_j \in V$ if such an edge exists, and zero otherwise. Further, we represent the labels as a label allocation matrix $L \in \mathbb{R}^{|\mathcal{L}|\times|V|}$ so that $L_{i,j} = 1$ if the j-th vertex has the i-th label and $L_{i,j} = 0$ otherwise. Because only a very small fraction of all the possible labels are ever assigned to any single node, this matrix is extremely sparse.

It is well known that when an adjacency matrix is multiplied with itself, each element $[A^2]_{i,j}$ contains the summed weight of paths from vertex v_i to vertex v_j through one intervening vertex, that is, paths of length two. Similarly, for any length n, the summed weights from v_i to v_j can be determined by calculating $[A^n]_{i,j}$.

Since we are interested not only in paths of one specific length, it is natural to combine the effect of paths of different lengths by summing the powers of the adjacency matrices. We calculate the infinite sum of the weights of all possible paths connecting

the vertices using the Neumann Series, defined as

$$(I - A)^{-1} = I + A + A^2 + ... = \sum_{k=0}^{\infty} A^k$$

if $|A| < 1$ where $|A|$ is the spectral radius of A (Meyer, 2000). From this sum we can form a new adjacency matrix

$$W = (I - A)^{-1} - I \ .$$

The final adjacency matrix contains the summed weights of all possible paths connecting the vertices. The identity matrix is subtracted to remove the paths of length zero, which would correspond to self-loops.

Next, we present the graph kernel that utilizes the graph representation defined previously. We define an instance G representing a candidate interaction as $G = LWL^{\mathrm{T}}$, where L and W are the label allocation matrix and the final adjacency matrix corresponding to the graph representation of the candidate interaction.

Following Gärtner et al. (2003) the graph kernel is defined as

$$k(G', G'') = \sum_{i=1}^{|\mathcal{L}|} \sum_{j=1}^{|\mathcal{L}|} G'_{i,j} G''_{i,j},$$

where G' and G'' are two instances formed as defined previously. The features can be thought as combinations of labels from connected pairs of vertices, with a value that represents the strength of their connection. In practical implementations, the full G matrices, which consist mostly of zeroes, are never explicitly formed. Rather, only the non-zero elements are stored in memory and used when calculating the kernels.

2.3 Scalable learning with Sparse RLS

RLS is a state-of-the-art kernel-based machine learning method which has been shown to have comparable performance to support vector machines (Rifkin et al., 2003). We choose the sparse version of the algorithm, also known as subset of regressors, as it allows us to scale up the method to very large training set sizes. Sparse RLS also has the property that it is possible to perform cross-validation and regularization parameter selection so that their time complexities are negligible compared to the training complexity. These efficient methods are analogous to the ones proposed by Pahikkala et al. (2006a) for the basic RLS regression.

We now briefly present the basic sparse RLS algorithm. Let m denote the training set size and $M = \{1, \ldots, m\}$ an index set in which the indices refer to the examples in the training set. Instead of allowing functions that can be expressed as a linear combination over the whole training set, as in the case of basic RLS regression, we only allow functions of the following restricted type:

$$f(\cdot) = \sum_{i \in B} a_i k(\cdot, x_i), \tag{1}$$

where k is the kernel function, x_i are training data points, $a_i \in \mathbb{R}$ are weights, and the set indexing the basis vectors $B \subset M$ is selected in advance. The coefficients a_i that determine (1) are obtained by minimizing

$$\sum_{i=1}^{m} (y_i - \sum_{j \in B} a_j k(x_i, x_j))^2 + \lambda \sum_{i,j \in B} a_i a_j k(x_i, x_j),$$

where the first term is the squared loss function, the second term is the regularizer, and $\lambda \in \mathbb{R}_+$ is a regularization parameter. Note that all the training instances are used for determining the coefficient vector. The minimizer is obtained by solving the corresponding system of linear equations, which can be performed in $O(m|B|^2)$ time.

We set the maximum number of basis vectors to 4000 in all experiments in this study. The subset is selected randomly when the training set size exceeds this number. Other methods for the selection of the basis vectors were considered by Rifkin et al. (2003), who however reported that the random selection worked as well as the more sophisticated approaches.

3 Experimental evaluation

We next describe the evaluation resources and metrics used, provide a comprehensive evaluation of our method across five PPI corpora, and compare our results to earlier work. Further, we discuss the challenges inherent in providing a valid method evaluation and propose solutions.

	Statistics		Graph Kernel						Co-occ.	
Corpus	#POS.	#NEG.	P	R	F	σ_F	AUC	σ_{AUC}	P	F
AIMed	1000	4834	0.529	0.618	0.564	0.050	0.848	0.023	0.178	0.301
BioInfer	1370	8924	0.477	0.599	0.529	0.053	0.849	0.065	0.135	0.237
HPRD50	163	270	0.643	0.658	0.634	0.114	0.797	0.063	0.389	0.554
IEPA	335	482	0.696	0.827	0.751	0.070	0.851	0.051	0.408	0.576
LLL	164	166	0.725	0.872	0.768	0.178	0.834	0.122	0.559	0.703

Table 1: Counts of positive and negative examples in the corpora and (P)recision, (R)ecall (F)-score and AUC for the graph kernel, with standard deviations provided for F and AUC.

3.1 Corpora and evaluation criteria

We evaluate our method using five publicly available corpora that contain PPI interaction annotation: AImed (Bunescu et al., 2005), BioInfer (Pyysalo et al., 2007a), HPRD50 (Fundel et al., 2007), IEPA (Ding et al., 2002) and LLL (Nédellec, 2005). All the corpora were processed to a common format using transformations[1] that we have introduced earlier (Pyysalo et al., 2008). We parse these corpora with the Charniak-Lease parser (Charniak and Lease, 2005), which has been found to perform best among a number of parsers tested in recent domain evaluations (Clegg and Shepherd, 2007; Pyysalo et al., 2007b). The Charniak-Lease phrase structure parses are transformed into the collapsed Stanford dependency scheme using the Stanford tools (de Marneffe et al., 2006). We cast the PPI extraction task as binary classification, where protein pairs that are stated to interact are positive examples and other co-occuring pairs negative. Thus, from each sentence, $\binom{n}{2}$ examples are generated, where n is the number of occurrences of protein names in the sentence. Finally, we form the graph representation described earlier for each candidate interaction.

We evaluate the method with 10-fold document-level cross-validation on all of the corpora. This guarantees the maximal use of the available data, and also allows comparison to relevant earlier work. In particular, on the AImed corpus we apply the exact same 10-fold split that was used by Bunescu et al. (2006) and Giuliano et al. (2006). Performance is measured according to the following criteria: interactions are considered untyped, undirected pairwise relations between specific protein mentions, that is, if the same protein name occurs multiple times in a sentence, the correct interactions must be extracted for each occurrence. Further, we do not consider self-interactions as candidates and remove them from the corpora prior to evaluation.

The majority of PPI extraction system evaluations use the balanced F-score measure for quantifying the performance of the systems. This metric is defined as $F = \frac{2pr}{p+r}$, where p is precision and r recall. Likewise, we provide F-score, precision, and recall values in our evaluation. It should be noted that F-score is very sensitive to the underlying positive/negative pair distribution of the corpus — a property whose impact on evaluation is discussed in detail below. As an alternative to F-score, we also evaluate the performance of our system using the *area under the receiver operating characteristics curve* (AUC) measure (Hanley and McNeil, 1982). AUC has the important property that it is invariant to the class distribution of the used dataset. Due to this and other beneficial properties for comparative evaluation, the usage of AUC for performance evaluation has been recently advocated in the machine learning community (see e.g. (Bradley, 1997)). Formally, AUC can be defined as

$$AUC = \frac{\sum_{i=1}^{m_+} \sum_{j=1}^{m_-} H(x_i - y_i)}{m_+ m_-},$$

where m_+ and m_- are the numbers of positive and negative examples, respectively, and x_1,...,x_{m_+} are the outputs of the system for the positive, and y_1,...,y_{m_-} for the negative examples, and

$$H(r) = \begin{cases} 1, & \text{if } r > 0 \\ 0.5, & \text{if } r = 0 \\ 0, & \text{otherwise.} \end{cases}$$

The measure corresponds to the probability that given a randomly chosen positive and negative ex-

[1] Available at `http://mars.cs.utu.fi/PPICorpora`.

ample, the system will be able to correctly disinguish which one is which.

3.2 Performance across corpora

The performance of our method on the five corpora for the various metrics is presented in Table 1. For reference, we show also the performance of the co-occurrence (or *all-true*) baseline, which simply assigns each candidate into the interaction class. The recall of the co-occurrence method is trivially 100%, and in terms of AUC it has a score of 0.5, the random baseline. All the numbers in Table 1 are averages taken over the ten folds. One should note that because of the non-linearity of the F-score measure, the average precision and recall will not produce exactly the average F.

The results hold several interesting findings. First, we briefly observe that on the AImed corpus, which has recently been applied in numerous evaluations (Sætre et al., 2008) and can be seen as an emerging *de facto* standard for PPI extraction method evaluation, the method achieves an F-score performance of 56.4%. As we argue in more detail below, this level of performance is comparable to the state-of-the-art in machine learning based PPI extraction. For the other large corpus, BioInfer, F-score performance is slightly lower.

Second, we observe that the F-score performance of the method varies strikingly between the different corpora, with results on IEPA and LLL approximately 20 percentage units higher than on AImed and BioInfer, despite the larger size of the latter two. In our previous work we have observed similar results with a rule-based extraction method (Pyysalo et al., 2008). As the first broad cross-corpus evaluation using a state-of-the-art machine learning method for PPI extraction, our results support and extend the key finding that F-score performance results measured on different corpora cannot, in general, be meaningfully compared.

The co-occurrence baseline numbers indicate one reason for the high F-score variance between the corpora. The F-score metric is not invariant to the distribution of positive and negative examples: for example, halving the number of negative test examples is expected to approximately halve the number of false positives at a given recall point. Thus, the greater the fraction of true interactions in a corpus is, the easier it is to reach high performance in terms of F-score. This is reflected in co-occurrence results, which range from 24% to 70% depending on the class distribution of the corpus.

This is a critical weakness of the F-score metric in cross-corpus comparisons as, for example, the fraction of true interactions out of all candidates is 50% on the LLL corpus but only 17% on AImed. By contrast to the large differences in performance measured using F-score, we find that for the distribution-invariant AUC measure the performance for all of the AImed, BioInfer, IEPA, and LLL corpora falls in the narrow range of 83-85%. In terms of AUC, performance on the HPRD50 corpus is an outlier, being approximately three percentage units lower than for any other corpus. Nevertheless, the results provide a strong argument in favor of applying the AUC metric instead of, or in addition to, F-score. AUC is also more stable in terms of variance.

Finally, we note that the similar performance in terms of AUC for corpora with as widely differing sizes as LLL and BioInfer indicates that past a relatively modest number of examples, increasing corpus size has a surprisingly small effect on the performance of the method. A similar finding can be seen, for example, in the relatively flat learning curve of Giuliano et al. (2006). While the issue requires further investigation, these results suggest that there may be more value in investing effort in developing better learning methods as opposed to larger corpora.

3.3 Performance compared to other methods

We next discuss the performance of our method compared to other methods introduced in the literature and the challenges of meaningful comparison, where we identify three major issues.

First, as indicated by the results above, differences in the makeup of different corpora render cross-corpus comparisons in terms of F-score essentially meaningless. As F-score is typically the only metric for which results are reported in the PPI extraction literature, we are limited to comparing against results on single corpora. We consider the AImed and BioInfer evaluations to be the most relevant ones, as these corpora are sufficiently large for training and reliably testing machine learning methods. As the present study is, to the best of our knowl-

	P	R	F
(Giuliano et al., 2006)	60.9%	57.2%	59.0%
All-dependency-paths graph kernel	52.9%	61.8%	56.4%
(Bunescu and Mooney, 2006)	65.0%	46.4%	54.2%
(Sætre et al., 2008)	64.3%	44.1%	52.0%
(Mitsumori et al., 2006)	54.2%	42.6%	47.7%
(Yakushiji et al., 2005)	33.7%	33.1%	33.4%

Table 2: (P)recision, (R)ecall and (F)-score results for methods evaluated on AImed with the correct cross-validation methodology.

edge, the first to report machine learning method performance on BioInfer, we will focus on AImed in the following comparison.

Second, the cross-validation strategy used in evaluation has a large impact on measured performance. In earlier system evaluations, two major strategies for defining the splits used in cross-validation can be observed. The approach used by Bunescu and Mooney (2006), which we consider the correct one, is to split the data into folds on level of documents. This guarantees that all pairs generated from the same document are always either in the training set or in the test set. Another approach is to pool all the generated pairs together, and then randomly split them to folds. To illustrate the significance of this choice, consider two interaction candidates extracted from the same sentence, e.g. from a statement of the form "P_1 and P_2 [...] P_3", where "[...]" is any statement of interaction or non-interaction. Due to the near-identity of contexts, a machine learning method will easily learn to predict that the label of the pair (P_1, P_2) should match that of (P_1, P_3). However, such "learning" will clearly not generalize. This approach must thus be considered invalid, because allowing pairs generated from same sentences to appear in different folds leads to an information leak between the training and test sets. Sætre et al. (2008) observed that adopting the latter cross-validation strategy on AImed could lead *up to 18 F-score percentage unit overestimation of performance*. For this reason, we will not consider results listed in the "False 10-fold cross-validation" table (2b) of Sætre et al. (2008).

With these restrictions in place, we now turn to comparison with relevant results reported in related research, summarized in Table 2. We note that Bunescu and Mooney (2006) only applied evaluation criteria where it is enough to extract only one occurrence of each mention of an interaction from each abstract, while the other results shown were evaluated using the same criteria as applied here. The former approach can produce higher performance: the evaluation of Giuliano et al. (2006) includes both alternatives, and their method achieves an F-score of 63.9% under the former criterion, which they term One Answer per Relation in a given Document (OARD). Our method outperforms most studies using similar evaluation methodology, with the exception being the approach of Giuliano et al. (2006). This result is somewhat surprising, as the method proposed by Giuliano does not apply any form of parsing but relies instead only on the sequential order of the words. This brings us to our third point regarding comparability of methods. As pointed out by Sætre et al. (2008), the AImed corpus allows remarkably different "interpretations" regarding the number of interacting and non-interacting pairs. For example, where we have identified 1000 interacting and 4834 non-interacting protein pairs in AImed, in the data used by Giuliano there are eight more interacting and 200 fewer non-interacting pairs. The corpus can also be preprocessed in a number of ways. In particular we noticed that whereas protein names are always blinded in our data, in the data used by Giuliano protein names are sometimes partly left visible. As Giuliano has generously made his method implementation available[2], we were able to test the performance of his system on the data we used in our experiments. This resulted in an F-score of 52.4%.

Finally, there remains an issue of parameter selection. For sparse RLS the values of the regular-

[2]Available at `http://tcc.itc.it/research/textec/tools-resources/jsre.html`.

ization parameter λ and the decision threshold separating the positive and negative classes must be chosen, which can be problematic when no separate data for choosing them is available. Choosing from several parameter values the ones that give best results in testing, or picking the best point from a precision/recall curve when evaluating in terms of F-score, will lead to an overoptimistic evaluation of performance. This issue has often not been addressed in earlier evaluations that do cross-validation on a whole corpus. We choose the parameters by doing further leave-one-document-out cross-validation within each round of 10-fold-cross-validation, on the nine folds that constitute the training set.

As a conclusion, we observe the results achieved with the all-dependency-paths kernel to be state-of-the-art level. However, differences in evaluation strategies and the large variance exhibited in the results make it impossible to state which of the systems considered can be expected in general to perform best. We encourage future PPI-system evaluations to report AUC and F-score results over multiple corpora, following clearly defined evaluation strategies, to bring further clarity to this issue.

4 Conclusions and future work

In this paper we have proposed a graph kernel approach to extracting protein-protein interactions, which captures the information in unrestricted dependency graphs to a format that kernel based learning algorithms can process. The method combines syntactic analysis with a representation of the linear order of the sentence, and considers all possible paths connecting any two vertices in the resulting graph. We demonstrate state-of-the art performance for the approach. All software developed in the course of this study is made publicly available at `http://mars.cs.utu.fi/PPICorpora`.

We identify a number of issues which make results achieved with different evaluation strategies and resources incomparable, or even incorrect. In our experimental design we consider the problems related to differences across corpora, the effects different cross-validation strategies have, and how parameter selection can be done. Our recommendation is to provide evaluations over different corpora, to use document-level cross-validation and to always selected parameters on the training set.

We draw attention to the behaviour of the F-score metric over corpora with differing pair distributions. The higher the relative frequency of interacting pairs is, the higher the performance can be expected to be. This is noticed both for the graph kernel method and for the naive co-occurrence baseline. Indeed, the strategy of just stating that all pairs interact leads to as high result as 70% F-score on one of the corpora. We consider AUC as an alternative measure that does not exhibit such behaviour, as it is invariant to the distribution of pairs. The AUC metric is much more stable across all the corpora, and never gives better results than random for approaches such as the naive co-occurrence.

Though we only consider binary interactions in this work, the graph representations have the property that they could be used to represent more complex structures than pairs. The availability of corpora that annotate complex interactions, such as the full BioInfer and GENIA, makes training a PPI extraction system for extracting complex interactions an important avenue of future research. However, how to avoid the combinatorial explosion following from considering triplets, quartets etc. remains an open question. Also, the performance of the current approaches may need to be yet improved before extending them to recognize complex interactions.

Acknowledgements

We would like to thank Razvan Bunescu, Claudio Giuliano and Rune Sætre for their generous assistance in providing us with data, software and information about their work on PPI extraction. Further, we thank CSC, the Finnish IT center for science, for providing us extensive computational resources. This work has been supported by the Academy of Finland and the Finnish Funding Agency for Technology and Innovation, Tekes.

References

Andrew P. Bradley. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159.

Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of HLT/EMNLP'05*, pages 724–731.

Razvan Bunescu and Raymond Mooney. 2006. Subsequence kernels for relation extraction. In *Proceedings of NIPS'05*, pages 171–178. MIT Press.

Razvan C. Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun Kumar Ramani, and Yuk Wah Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artif Intell Med*, 33(2):139–155.

Eugene Charniak and Matthew Lease. 2005. Parsing biomedical literature. In *Proceedings of IJCNLP'05*, pages 58–69.

Andrew Brian Clegg and Adrian Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8(1):24.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC'06*, pages 449–454.

J. Ding, D. Berleant, D. Nettleton, and E. Wurtele. 2002. Mining MEDLINE: abstracts, sentences, or phrases? In *Proceedings of PSB'02*, pages 326–337.

Katrin Fundel, Robert Kuffner, and Ralf Zimmer. 2007. RelEx–Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Thomas Gärtner, Peter A. Flach, and Stefan Wrobel. 2003. On graph kernels: Hardness results and efficient alternatives. In *COLT'03*, pages 129–143. Springer.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of EACL'06*.

James A. Hanley and B. J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Lawrence Hunter, Zhiyong Lu, James Firby, William A. Baumgartner, Helen L Johnson, Philip V. Ogren, and K. Bretonnel Cohen. 2008. OpenDMAP: An open-source, ontology-driven concept analysis engine, with applications to capturing knowledge regarding protein transport, protein interactions and cell-specific gene expression. *BMC Bioinformatics*, 9(78).

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(10).

Carl D. Meyer. 2000. *Matrix analysis and applied linear algebra*. Society for Industrial and Applied Mathematics.

Tomohiro Mitsumori, Masaki Murata, Yasushi Fukuda, Kouichi Doi, and Hirohumi Doi. 2006. Extracting protein-protein interaction information from biomedical text with svm. *IEICE - Trans. Inf. Syst.*, E89-D(8):2464–2466.

Claire Nédellec. 2005. Learning language in logic - genic interaction extraction challenge. In *Proceedings of LLL'05*.

Tapio Pahikkala, Jorma Boberg, and Tapio Salakoski. 2006a. Fast n-fold cross-validation for regularized least-squares. In *Proceedings of SCAI'06*, pages 83–90.

Tapio Pahikkala, Evgeni Tsivtsivadze, Jorma Boberg, and Tapio Salakoski. 2006b. Graph kernels versus graph representations: a case study in parse ranking. In *Proceedings of the ECML/PKDD'06 workshop on Mining and Learning with Graphs*.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007a. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50).

Sampo Pyysalo, Filip Ginter, Veronika Laippala, Katri Haverinen, Juho Heimonen, and Tapio Salakoski. 2007b. On the unification of syntactic annotations under the stanford dependency scheme: A case study on BioInfer and GENIA. In *Proceedings of BioNLP'07*, pages 25–32.

Sampo Pyysalo, Antti Airola, Juho Heimonen, Jari Björne, Filip Ginter, and Tapio Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics, special issue*, 9(Suppl 3):S6.

Ryan Rifkin, Gene Yeo, and Tomaso Poggio, 2003. *Regularized Least-squares Classification*, volume 190 of *NATO Science Series III: Computer and System Sciences*, chapter 7, pages 131–154. IOS Press.

Rune Sætre, Kenji Sagae, and Jun'ichi Tsujii. 2008. Syntactic features for protein-protein interaction extraction. In *Proceedings of LBM'07*, volume 319, pages 6.1–6.14.

Akane Yakushiji, Yusuke Miyao, Yuka Tateisi, and Jun'ichi Tsujii. 2005. Biomedical information extraction with predicate-argument structure patterns. In *Proceedings of SMBM'05*, pages 60–69.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res.*, 3:1083–1106.

Extracting Clinical Relationships from Patient Narratives

Angus Roberts, Robert Gaizauskas, Mark Hepple
Department of Computer Science, University of Sheffield,
Regent Court, 211 Portobello, Sheffield S1 4DP
{initial.surname}@dcs.shef.ac.uk

Abstract

The Clinical E-Science Framework (CLEF) project has built a system to extract clinically significant information from the textual component of medical records, for clinical research, evidence-based healthcare and genotype-meets-phenotype informatics. One part of this system is the identification of relationships between clinically important entities in the text. Typical approaches to relationship extraction in this domain have used full parses, domain-specific grammars, and large knowledge bases encoding domain knowledge. In other areas of biomedical NLP, statistical machine learning approaches are now routinely applied to relationship extraction. We report on the novel application of these statistical techniques to clinical relationships.

We describe a supervised machine learning system, trained with a corpus of oncology narratives hand-annotated with clinically important relationships. Various shallow features are extracted from these texts, and used to train statistical classifiers. We compare the suitability of these features for clinical relationship extraction, how extraction varies between inter- and intra-sentential relationships, and examine the amount of training data needed to learn various relationships.

1 Introduction

The application of Natural Language Processing (NLP) is widespread in biomedicine. Typically, it is applied to improve access to the ever-burgeoning research literature. Increasingly, biomedical researchers need to relate this literature to phenotypic data: both to populations, and to individual clinical subjects. The computer applications used in biomedical research, including NLP applications, therefore need to support genotype-meets-phenotype informatics and the move towards translational biology. Such support will undoubtedly include linkage to the information held in individual medical records: both the structured portion, and the unstructured textual portion.

The Clinical E-Science Framework (CLEF) project (Rector et al., 2003) is building a framework for the capture, integration and presentation of this clinical information, for research and evidence-based health care. The project's data resource is a repository of the full clinical records for over 20000 cancer patients from the Royal Marsden Hospital, Europe's largest oncology centre. These records combine structured information, clinical narratives, and free text investigation reports. CLEF uses information extraction (IE) technology to make information from the textual portion of the medical record available for integration with the structured record, and thus available for clinical care and research. The CLEF IE system analyses the textual records to extract entities, events and the relationships between them. These relationships give information that is often not available in the structured record. Why was a drug given? What were the results of a physical examination? What problems were not present? We have previously reported entity extraction in the CLEF IE system (Roberts et al., 2008b). This paper examines relationship extraction.

Extraction of relationships from clinical text is usually carried out as part of a full clinical IE system. Several such systems have been described. They generally use a syntactic parse with domain-specific grammar rules. The Linguistic String project (Sager et al., 1994) used a full syntactic and

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 10–18,
Columbus, Ohio, USA, June 2008.

clinical sublanguage parse to fill template data structures corresponding to medical statements. These were mapped to a database model incorporating medical facts and the relationships between them. MedLEE (Friedman et al., 1994), and more recently BioMedLEE (Lussier et al., 2006) used a semantic lexicon and grammar of domain-specific semantic patterns. The patterns encode the possible relationships between entities, allowing both entities and the relationships between them to be directly matched in the text. Other systems have incorporated large-scale domain-specific knowledge bases. MEDSYNDIKATE (Hahn et al., 2002) employed a rich discourse model of entities and their relationships, built using a dependency parse of texts and a description logic knowledge base re-engineered from existing terminologies. MENELAS (Zweigenbaum et al., 1995) also used a full parse, a conceptual representation of the text, and a large scale knowledge base.

In other applications of biomedical NLP, a second paradigm has become widespread: the application of statistical machine learning techniques to feature-based models of the text. Such approaches have typically been applied to journal texts. They have been used both for entity recognition and extraction of various relations, such as protein-protein interactions (see, for example, Grover et al (2007)). This follows on from the success of these methods in general NLP (see for example Zhou et al (2005)). Statistical machine learning has also been applied to clinical text, but its use has generally been limited to entity recognition. The Mayo Clinic text analysis system (Pakhomov et al., 2005), for example, uses a combination of dictionary lookup and a Naïve Bayes classifier to identify entities for information retrieval applications. To the best of our knowledge, statistical methods have not been previously applied to extraction of clinical relationships from text.

This paper describes experiments in the statistical machine learning of relationships from a novel text type: oncology narratives. The set of relationships extracted are considered to be of interest for clinical and research applications down line of IE, such as querying to support clinical research. We apply Support Vector Machine (SVM) classifiers to learn these relationships. The classifiers are trained and evaluated using novel data: a gold standard corpus of clinical text, hand-annotated with semantic entities and relationships. In order to test the applicability of this method to the clinical domain, we train classifiers using a number of comparatively simple text features, and look at the contribution of these features to system performance. Clinically interesting relationships may span several sentences, and so we compare classifiers trained for both intra- and inter-sentential relationships (spanning one or more sentence boundaries). We also examine the influence of training corpus size on performance, as hand annotation of training data is the major expense in supervised machine learning.

2 Relationship Schema

Relationship	Argument 1	Argument 2
has_target	Investigation	Locus
	Intervention	Locus
has_finding	Investigation	Condition
	Investigation	Result
has_indication	Drug or device	Condition
	Intervention	Condition
	Investigation	Condition
has_location	Condition	Locus
negation_modifies	Negation modifier	Condition
laterality_modifies	Laterality modifier	Intervention
	Laterality modifier	Locus
sub-location_modifies	Sub-location modifier	Locus

Table 1: Relationship types and their argument type constraints.

The CLEF application extracts entities, relationships and modifiers from text. By *entity*, we mean some real-world thing, event or state referred to in the text: the drugs that are mentioned, the tests that were carried out, etc. *Modifiers* are words that qualify an entity in some way, referring e.g. to the laterality of an anatomical locus, or the negation of a condition ("no sign of inflammation"). Entities are connected to each other and to modifiers by *relationships*: e.g. linking a drug entity to the condition entity for which it is indicated, linking an investigation to its results, or linking a negating phrase to a condition.

The entities, modifiers, and relationships are described by both a formal XML schema, and by a set of detailed definitions. These were developed by a group of clinical experts through an iterative process, until acceptable agreement was reached. Entity types are mapped to types from the UMLS semantic network (Lindberg et al., 1993), each CLEF en-

tity type covering several UMLS types. Relationship types are those that were felt necessary to capture the essential clinical dependencies between entities referred to in patient documents, and to support CLEF end user applications.

Each relationship type is constrained to exist between limited pairs of entity types. For example, the `has_location` relationship can only exist between a `Condition` entity and a `Locus` entity. Some relationships can exist between multiple type pairs. The full set of relationships and their argument type constraints are shown in Table 1. Examples of each relationship are given in Roberts et al (2008a).

Some of the relationships considered important by the clinical experts were not obvious without domain knowledge. For example,

> He is suffering from nausea and severe headaches. Dolasteron was prescribed.

Without domain knowledge, it is not clear that there is a `has_indication` relationship between the "Dolasteron" `Drug or device` entity and the "nausea" `Condition` entity. As in this example, many of this type of relationship are intra-sentential.

A single real-world entity may be referred to several times in the same text. Each of these co-referring expressions is a *mention* of the entity. The gold standard includes annotation of co-reference between different textual mentions of the same entity. For the work reported in this paper, however, co-reference is not considered. Each entity is assumed to have a single mention. Relationships between entities can be considered, by extension, as relationships between the single mentions of those entities. The implications of this are discussed further below.

3 Gold Standard Corpus

The schema and definitions were used to hand-annotate the entities and relationships in 77 oncology narratives, to provide a gold standard for system training and evaluation. Corpora of this size are typical in supervised machine learning, and reflect the expense of hand annotation. Narratives were carefully selected and annotated according to a best practice methodology, as described in Roberts et al (2008a). Narratives were annotated by two independent, clinically trained, annotators, and a consensus created by a third. We will refer to this corpus as C77.

Annotators were asked to first mark the mentions of entities and modifiers, and then to go through each of these in turn, deciding if any had relationships with mentions of other entities. Although the annotators were marking co-reference between mentions of the same entity, they were asked to ignore this with respect to relationship annotation. Both the annotation tool that they were using and their annotation guidelines, enforced the creation of relationships between mentions, and not between entities. The gold standard is thus analogous to the style of relationship extraction reported here, in which we extract relations between single mention entities, and do not consider co-reference. Annotators were further told that relationships could span multiple sentences, and that it was acceptable to use clinical domain knowledge to infer that a relationship existed between two mentions. Counts of all relationships annotated in C77 are shown in Table 2, sub-divided by the number of sentence boundaries spanned by a relationship.

4 Relationship Extraction

The system we have built uses the GATE NLP toolkit (Cunningham et al., 2002) [1]. The system is shown in Figure 1, and is described below.

Narratives are first pre-processed using standard GATE modules. Narratives were tokenised, sentences found with a regular expression-based sentence splitter, part-of-speech (POS) tagged, and morphological roots found for tokens. Each token was also labelled with a generalised POS tag, the first two characters of the full POS tag. This takes advantage of the Penn Treebank tagset used by GATE's POS tagger, in which related POS tags share the first two characters. For example, all six verb POS tags start with the letters "VB".

After pre-processing, mentions of entities within the text are annotated. In the experiments reported, we assume perfect entity recognition, as given by the entities in the human annotated gold standard

[1] We used a development build of GATE 4.0, downloadable from `http://gate.ac.uk`

	Sentence boundaries between arguments											Total
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**>9**	
has_finding	265	46	25	7	5	4	3	2	2	2	0	361
has_indication	139	85	35	32	14	11	6	4	5	5	12	348
has_location	360	4	1	1	1	1	1	0	0	0	4	373
has_target	122	14	4	2	2	4	3	1	0	1	0	153
laterality_modifies	128	0	0	0	0	0	0	0	0	0	0	128
negation_modifies	100	1	0	0	0	0	0	0	0	0	0	101
sub_location_modifies	76	0	0	0	0	0	0	0	0	0	0	76
Total	1190	150	65	42	22	20	13	7	7	8	16	1540
Cumulative total	1190	1340	1405	1447	1469	1489	1502	1509	1516	1524	1540	

Table 2: Count of relationships in 77 gold standard documents.

described above. Our results are therefore higher than would be expected in a system with automatic entity recognition. It is useful and usual to fix entity recognition in this way, to allow tuning specific to relationship extraction, and to allow the isolation of relation-specific problems. We accept, however, that ultimately, relation extraction does depend on the quality of entity recognition. The relation extraction described here is used as part of an operational IE system in which clinical entity recognition is performed by a combination of lexical lookup and supervised machine learning. We have described our entity extraction system elsewhere (Roberts et al., 2008b).

4.1 Classification

We treat clinical relationship extraction as a classification task, training classifiers to assign a relationship type to an *entity pair*. An entity pair is a pairing of entities that may or may not be the arguments of a relation. For a given document, we create all possible entity pairs within two constraints. First, entities that are paired must be within n sentences of each other. For all of the work reported here, unless stated, $n \leq 1$ (crossing 0 or 1 sentence boundaries). Second, we can constrain the entity pairs created by argument type (Rindflesch and Fiszman, 2003). For example, there is little point in creating an entity pair between a `Drug or device` entity and a `Result` entity, as no relationships, as specified by the schema, exist between entities of these types. Entity pairing is carried out by a GATE component developed specifically for clinical relationship extraction. In addition to pairing entities according to the above constraints, this component also assigns features to each pair that characterise its lexical and syntactic qualities (described further in Section 4.2).

Entity pairs correspond to classifier training and test instances. In classifier training, if an entity pair corresponds to the arguments of a relationship present in the gold standard, then it is assigned a class of that relationship type. If it does not correspond to such a relation, then it is assigned the class `null`. The classifier builds a model of these entity pair training instances, from their features. In classifier application, entity pairs are created from unseen text, under the above constraints. The classifier assigns one of our seven relationship types, or `null`, to each entity pair.

We use Support Vector machines (SVMs) as trainable classifiers, as these have proved to be robust and efficient for a range of NLP tasks, including relation extraction. We use an SVM implementation developed within our own group, and provided as part of the GATE toolkit. This is a variant on the original SVM algorithm, SVM with uneven margins, in which classification may be biased towards positive training examples. This is particularly suited to NLP applications, in which positive training examples are often rare. Full details of the classifier are given in Li et al (2005). We used the implementation "out of the box", with default parameters as determined in experiments with other data sets.

SVMs are binary classifiers: the multi-class problem of classifying entity pairs must therefore be mapped to a number of binary classification problems. There are several ways in which a multi-class problem can be recast as binary problems. The commonest are *one-against-one* in which one classifier is trained for every possible pair of classes, and *one-against-all* in which a classifier is trained for a binary decision between each class and all other

classes, including `null`, combined. We have carried out extensive experiments (not reported here), with these two strategies, and have found little difference between them for our data. We have chosen to use one-against-all, as it needs fewer classifiers (for an n class problem, it needs n classifiers, as opposed to $\frac{(n-1)!}{2}$ for one-against-one).

The resultant class assignments by multiple binary classifiers must be post-processed to deal with ambiguity. In application to unseen text, it is possible that several classifiers assign different classes to an entity pair (test instance). To disambiguate these cases, the output of each one-against-all classifier is transformed into a probability, and the class with the highest probability is assigned. Re-casting the multi-class relation problem as a number of binary problems, and post-processing to resolve ambiguities, is handled by the GATE Learning API.

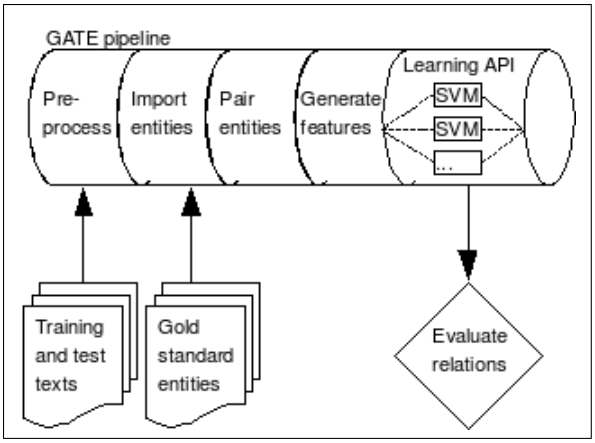

Figure 1: The relationship extraction system.

4.2 Features for Classification

The SVM classification model is built from lexical and syntactic features assigned to tokens and entity pairs prior to classification. We use features developed in part from those described in Zhou et al (2005) and Wang et al (2006). These features are split into 11 sets, as described in Table 3.

The `tokN` features are POS and surface string taken from a window of N tokens on each side of each paired entity's mention. For $N = 6$, this gives 48 features. The rationale behind these simple features is that there is useful information in the words surrounding two mentions, that helps determine any relationship between them. The `gentokN` features generalise `tokN` to use morphological root and generalised POS. The `str` features are a set of 14 surface string features, encoding the full surface strings of both entity mentions, their heads, their heads combined, the surface strings of the first, last and other tokens between the mentions, and of the two tokens immediately before and after the leftmost and rightmost mentions respectively. The `pos`, `root`, and `genpos` feature sets are similarly constructed from the POS tags, roots, and generalised POS tags of the entity mentions and their surrounding tokens. These four feature sets differ from `tokN` and `gentokN`, in that they provide more fine-grained information about the position of features relative to the paired entity mentions.

For the `event` feature set, the main entities were divided into events (`Investigation` and `Intervention`) and non-events (all others). Features record whether the entity pair consists of two events, two non-events, one of each, and whether there are any intervening events and non-events. This feature set gives similar information to `atype` (semantic types of arguments) and `inter` (intervening entities), but at a coarser level of typing.

5 Evaluation

We used a standard ten-fold cross validation methodology and standard evaluation metrics. Metrics are defined in terms of true positive, false positive and false negative matches between relationships in a system annotated *response* document and a gold standard *key* document. A response relationship is a true positive if a relationship of the same type, and with the exact same arguments, exists in the key. Corresponding definitions apply for false positive and false negative. Counts of these matches are used to calculate standard metrics of Recall (R), Precision (P) and $F1$ measure.

The metrics do not say how hard relationship extraction is. We therefore provide a comparison with Inter Annotator Agreement (IAA) scores from the gold standard. The IAA score gives the agreement between the two independent double annotators. It is equivalent to scoring one annotator against the other using the $F1$ metric. IAA scores are not directly comparable here, as relationship annotation is

Feature set	Size	Description
tokN	$8N$	Surface string and POS of tokens surrounding the arguments, windowed $-N$ to $+N$, $N = 6$ by default
gentokN	$8N$	Root and gerenalised POS of tokens surrounding the argument entities, windowed $-N$ to $+N$, $N = 6$ by default
atype	1	Concatenated semantic type of arguments, in arg1-arg2 order
dir	1	Direction: linear text order of the arguments (is arg1 before arg2, or vice versa?)
dist	2	Distance: absolute number of sentence and paragraph boundaries between arguments
str	14	Surface string features based on Zhou et al (2005), see text for full description
pos	14	POS features, as above
root	14	Root features, as above
genpos	14	Generalised POS features, as above
inter	11	Intervening mentions: numbers and types of intervening entity mentions between arguments
event	5	Events: are any of the arguments, or intevening entities, events?
allgen	96	All features in root and generalised POS forms, i.e. gentok6+atype+dir+dist+root+genpos+inter+event
notok	48	All except tokN features, others in string and POS forms, i.e. atype+dir+dist+str+pos+inter+event

Table 3: Feature sets used for learning relationships. The size of a set is the number of features in that set.

a slightly different task for the human annotators. The relationship extraction system is given entities, and finds relationships between them. Human annotators must find both the entities and the relationships. Therefore, were one human annotator to fail to find a particular entity, they could never find relationships with that entity. The raw IAA score does not take this into account: if an annotator fails to find an entity, then they will also be penalised for all relationships with that entity. We therefore give a Corrected IAA, CIAA, in which annotators are only compared on those relations for which they have both found the entities involved. Both forms of IAA are shown in Table 4. It is clear that it is hard for annotators to reach agreement on relationships, and that this is compounded massively by lack of perfect agreement on entities. Note that the gold standard used in training and evaluation reflects a further consensus annotation, to correct this poor agreement.

6 Results

6.1 Feature Selection

The first group of experiments reported looks at the performance of relation extraction with various feature sets. We followed an additive strategy for feature selection. Starting with basic features, we added further features one set at a time. We measured the performance of the resulting classifier each time we added a new feature set. Results are shown in Table 4. The initial classifier used a `tok6+atype` feature set. Addition of both `dir` and `dist` features give significant improvements in all metrics, of around 10% $F1$ overall, in each case. This suggests that the linear text order of arguments, and whether relations are intra- or inter-sentential is important to classification. Addition of the `str` features also give good improvement in most metrics, again 10% $F1$ overall. Addition of part-of-speech information, in the form of `pos` features, however, leads to a drop in some metrics, overall $F1$ dropping by 1%. Unexpectedly, POS seems to provide little extra information above that in surface string. Errors in POS tagging cannot be dismissed, and could be the cause of this. The existence of intervening entities, as coded in feature set `inter`, provides a small benefit. The inclusion of information about events, in the `event` feature set, is less clear-cut.

We were interested to see if generalising features could improve performance, as this had benefited our previous work in entity extraction. We replaced all surface string features with their root form, and POS features with their generalised POS form. This gave the results shown in column `allgen`. Results are not clear cut, in some cases better and in some worse than the previous best. Overall, there is no difference in $F1$. There is a slight increase in overall recall, and a corresponding drop in precision — as might be expected.

Both the `tokN`, and the `str` and `pos` feature sets provide surface string and POS information about tokens surrounding and between relationship arguments. The former gives features from a window around each argument. The latter two give a greater amount of positional information. Do these two provide enough information on their own, without the windowed features? To test this, we removed the `tokN` features from the full cumulative feature set, from column `+event`. Results are given in column

Relation	Metric	tok6+atype	+dir	+dist	+str	+pos	+inter	+event	allgen	notok	IAA	CIAA
has_finding	**P**	44	49	58	63	62	64	65	63	63		
	R	39	63	78	80	80	81	81	82	82		
	F1	39	54	66	70	69	71	72	71	71	46	80
has_indication	**P**	37	23	38	42	40	41	42	37	44		
	R	14	14	46	44	44	47	47	45	47		
	F1	18	16	39	39	38	41	42	38	41	26	50
has_location	**P**	36	36	50	68	71	72	72	73	73		
	R	28	28	74	79	79	81	81	83	83		
	F1	30	30	58	72	74	76	75	77	76	55	80
has_target	**P**	9	9	32	63	57	60	62	60	59		
	R	11	11	51	68	67	67	66	68	68		
	F1	9	9	38	64	60	63	63	63	62	42	63
laterality_modifies	**P**	21	38	73	84	83	84	84	86	86		
	R	9	55	82	89	86	88	88	87	89		
	F1	12	44	76	85	83	84	84	84	85	73	94
negation_modifies	**P**	19	54	85	81	80	79	79	77	81		
	R	12	82	97	98	93	92	93	93	93		
	F1	13	63	89	88	85	84	85	83	85	66	93
sub_location_modifies	**P**	2	2	55	88	86	86	88	88	87		
	R	1	1	62	94	92	95	95	95	95		
	F1	1	1	56	90	86	89	91	91	90	49	96
Overall	**P**	33	38	50	63	62	64	65	64	64		
	R	22	36	70	74	73	75	75	76	76		
	F1	26	37	58	68	67	69	69	69	70	47	75

Table 4: Variation in performance by feature set. Features sets are abbreviated as in Table 3. For the first seven columns, features were added cumulatively to each other. The next two columns, `allgen` and `notok`, are as described in Table 3. The final two columns give inter annotator agreement and corrected inter annotator agreement, for comparison.

`notok`. There is no clear change in performance, some relationships improving, and some worsening. Overall, there is a 1% improvement in $F1$.

It appears that the bulk of performance is attained through entity type and distance features, with some contribution from positional surface string information. Performance is between 1% and 9% lower than CIAA for the same relationship, with a best overall $F1$ of 70%, compared to a CIAA of 75%.

6.2 Sentences Spanned

Table 2 shows that although most relationships are intra-sentential, 23% are inter-sentential, 10% of all relationships being between arguments in adjacent sentences. If we consider a relationship to cross n sentence boundaries, then the classifiers described in the previous section were all trained on relationships crossing $n \leq 1$ sentence boundaries, i.e. with arguments in the same or adjacent sentences. What effect does including more distant relationships have on performance? We trained classifiers on only intra-sentential relationships, and on relationships spanning up to n sentence boundaries, for $n \in \{1...5\}$. We also trained a classifier on relationships with $1 \leq n \leq 5$, comprising 85% of all inter-sentential relationships. In each case, the cumulative feature set `+event` from Table 4 was used. Results are shown in Table 5. It is clear from the results that the feature sets used do not perform well on inter-sentential relationships. There is a 6% drop in overall $F1$ when including relationships with $n = 1$ together with $n < 1$. Performance continues to drop as more inter-sentential relationships are included, and is very poor for just inter-sentential relationships.

A preliminary error analysis suggests that the more distant relationship arguments are from each other, the more likely clinical knowledge is required to extract the relationship. This raises additional difficulties for extraction, which the simple features described here are unable to address.

6.3 Size of Training Corpus

The provision of sufficient training data for supervised learning algorithms is a limitation on their use. We examined the effect of training corpus size on relationship extraction. The C77 corpus, compris-

		Number of sentence boundaries between arguments							Corpus size		
		inter-	intra-	inter- and intra-sentential							
Relation	**Metric**	$\mathbf{1 \leq n \leq 5}$	$\mathbf{n < 1}$	$\mathbf{n \leq 1}$	$\mathbf{n \leq 2}$	$\mathbf{n \leq 3}$	$\mathbf{n \leq 4}$	$\mathbf{n \leq 5}$	**C25**	**C50**	**C77**
has_finding	**P**	24	68	65	62	60	61	61	66	63	65
	R	18	89	81	79	78	78	77	74	74	81
	F1	18	76	72	69	67	68	67	67	67	72
has_indication	**P**	18	49	42	42	36	32	30	22	25	42
	R	17	59	47	42	42	39	38	30	31	47
	F1	16	51	42	39	37	34	33	23	25	42
has_location	**P**	0	74	72	73	72	72	72	72	71	72
	R	0	83	81	81	81	82	82	76	80	81
	F1	0	77	75	76	75	76	76	73	74	75
has_target	**P**	3	64	62	59	60	59	58	65	49	62
	R	1	75	66	64	62	61	61	60	65	66
	F1	2	68	63	61	60	60	59	59	54	63
laterality_modifies	**P**	0	86	84	86	86	86	87	77	78	84
	R	0	89	88	88	88	87	88	69	68	88
	F1	0	85	84	85	86	85	86	72	69	84
negation_modifies	**P**	0	80	79	79	80	80	80	78	79	79
	R	0	94	93	91	93	93	93	80	93	93
	F1	0	86	85	84	85	86	85	78	84	85
sub_location_modifies	**P**	0	89	88	88	89	89	89	64	91	88
	R	0	95	95	95	95	95	95	64	85	95
	F1	0	91	91	91	91	91	91	64	86	91
Overall	**P**	22	69	65	64	62	61	60	62	63	65
	R	17	83	75	73	71	70	70	65	71	75
	F1	19	75	69	68	66	65	65	63	66	69

Table 5: Variation in performance, by number of sentence boundaries (n), and by training corpus size.

ing 77 narratives and used in the previous experiments, was subsetted to give corpora of 25 and 50 narratives, which will be referred to as C25 and C50 respectively. We trained two further classifiers on these new corpora. Again, the cumulative feature set `+event` from Table 4 was used. Results are shown in Table 5. Overall, performance improves as training corpus size increases ($F1$ rising from 63% to 69%). We were struck however, by the fact that increasing from 50 to 77 documents has little effect on a few relationships (`negation_modifies` and `has_location`). It may well be that the amount of training data required has plateaued for those relationships.

7 Conclusion

We have shown that it is possible to extract clinical relationships from text, using shallow features, and supervised statistical machine learning. Judging from poor inter annotator agreement, the task is hard. Our system achieves a reasonable performance, with an overall $F1$ just 5% below a corrected inter annotator agreement. This performance is reached largely by using features of the text that encode entity type, distance between arguments, and some surface string information. Performance does, however, vary with the number of sentences spanned by the relationships. Learning inter-sentential relationships does not seem amenable to this approach, and may require the use of domain knowledge.

A major concern when using supervised learning algorithms is the expense and availability of training data. We have shown that while this concern is justified in some cases, larger training corpora may not improve performance for all relationships.

The technology used has proved scalable. The full CLEF IE system, including automatic entity recognition, is able to process a document in subsecond time on a commodity workstation. We have used the system to extract 6 million relations from over half a million patient documents, for use in downstream CLEF applications (Roberts et al., 2008a). Our future work on relationship extraction in CLEF includes integration of a dependency parse into the feature set, further analysis to determine what knowledge may be required to learn intersentential relations, and integration of relationship extraction with a co-reference algorithm.

Availability All of the software described here is open source and can be downloaded as part of GATE, with the exception of the entity pairing component, which will be released shortly. We are currently preparing a UK research ethics committee application, requesting permission to release our annotated corpus.

Acknowledgements

CLEF is funded by the UK Medical Research Council. We would like to thank the Royal Marsden Hospital for providing the corpus, and our clinical partners in CLEF for assistance in developing the schema, and for gold standard annotation.

References

H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*, pages 168–175, Philadelphia, PA, USA, July.

C. Friedman, P. Alderson, J. Austin, J. Cimino, and S. Johnson. 1994. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174, March.

C. Grover, B. Haddow, E. Klein, M. Matthews, L. Nielsen, R. Tobin, and X. Wang. 2007. Adapting a relation extraction pipeline for the BioCreAtIvE II task. In *Proceedings of the BioCreAtIvE II Workshop 2007*, Madrid, Spain.

U. Hahn, M. Romacker, and S. Schulz. 2002. MEDSYNDIKATE — a natural language system for the extraction of medical information from findings reports. *International Journal of Medical Informatics*, 67(1–3):63–74, December.

Y. Li, K. Bontcheva, and H. Cunningham. 2005. SVM based learning system for information extraction. In *Deterministic and statistical methods in machine learning: first international workshop*, number 3635 in Lecture Notes in Computer Science, pages 319–339. Springer.

D. Lindberg, B. Humphreys, and A. McCray. 1993. The Unified Medical Language System. *Methods of Information in Medicine*, 32(4):281–291.

Y. Lussier, T. Borlawsky, D. Rappaport, Y. Liu, and C. Friedman. 2006. PhenoGO: Assigning phenotypic context to Gene Ontology annotations with natural language processing. In *Biocomputing 2006, Proceedings of the Pacific Symposium*, pages 64–75, Hawaii, USA, January.

S. Pakhomov, J. Buntrock, and P. Duffy. 2005. High throughput modularized NLP system for clinical text. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), interactive poster and demonstration sessions*, pages 25–28, Ann Arbor, MI, USA, June.

A. Rector, J. Rogers, A. Taweel, D. Ingram, D. Kalra, J. Milan, P. Singleton, R. Gaizauskas, M. Hepple, D. Scott, and R. Power. 2003. CLEF — joining up healthcare with clinical and post-genomic research. In *Proceedings of UK e-Science All Hands Meeting 2003*, pages 264–267, Nottingham, UK.

T. Rindflesch and M. Fiszman. 2003. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of Biomedical Informatics*, 36(6):462–477.

A. Roberts, R. Gaizauskas, M. Hepple, G. Demetriou, Y. Guo, A. Setzer, and I. Roberts. 2008a. Semantic annotation of clinical text: The CLEF corpus. In *Proceedings of Building and evaluating resources for biomedical text mining: workshop at LREC 2008*, Marrakech, Morocco, May. In press.

A. Roberts, R. Gaizauskas, M. Hepple, and Y. Guo. 2008b. Combining terminology resources and statistical methods for entity recognition: an evaluation. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2008*, Marrakech, Morocco, May. In press.

N. Sager, M. Lyman, C. Bucknall, N. Nhan, and L. Tick. 1994. Natural language processing and the representation of clinical data. *Journal of the American Medical Informatics Association*, 1(2):142–160, March-April.

T. Wang, Y. Li, K. Bontcheva, H. Cunningham, and J. Wang. 2006. Automatic extraction of hierarchical relations from text. In *The Semantic Web: Research and Applications. 3rd European Semantic Web Conference, ESWC 2006*, number 4011 in Lecture Notes in Computer Science, pages 215–229. Springer.

G. Zhou, J. Su, J. Zhang, and M. Zhang. 2005. Exploring Various Knowledge in Relation Extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 427–434, Ann Arbor, MI, USA, June.

P. Zweigenbaum, B. Bachimont, J. Bouaud, J. Charlet, and J-F. Boisvieux. 1995. A multi-lingual architecture for building a normalised conceptual representation from medical language. In *Proceedings of the Annual Symposium on Computer Applications in Medical Care*, pages 357–361, New York, NY, USA.

Using Automated Feature Optimisation to Create an Adaptable Relation Extraction System

Barry Haddow
School of Informatics, University of Edinburgh,
2 Buccleuch Place, Edinburgh, Scotland, EH8 9LW
bhaddow@inf.ed.ac.uk

Abstract

An adaptable relation extraction system for the biomedical domain is presented. The system makes use of a large set of contextual and shallow syntactic features, which can be automatically optimised for each relation type. The system is tested on three different relation types; protein-protein interactions, tissue expression relations and fragment to parent protein relations.

1 Introduction

In biomedical information extraction, research in named entity recognition (NER) and relation extraction (RE) has tended to focus on the extracting proteins and their interactions, with less thought given to how to adapt such systems to other entities and relations of biomedical interest. This is especially true for RE, where there is very little work on relations other than protein-protein interactions. Nevertheless, in order to create applications of use to biologists such as curation assistants and improved information extraction and retrieval systems it will be necessary to treat a broader range of semantic relations. The recent release of the Genia event corpus (Kim et al., 2008) will help to drive this research.

The aim of this paper is to address the problem of how to create an RE system, which can be adapted to different biomedical RE problems with a minimum of manual intervention. Since this paper focuses on relation extraction, it will be assumed that the named entities are given, in other words the human annotated entities are used in all experiments. The approach taken to RE is to treat it as a supervised classification problem on relation candidates, using a large collection of shallow syntactic and contextual features. Relation candidates are pairs of entities, picked out using an appropriate candidate generation strategy. The use of shallow (as opposed to deep) syntactic features means that the system can rely on relatively robust linguistic tools such as part-of-speech taggers and chunkers, rather than more brittle and less widely available tools such as parsers. The difficulty with feature-based methods is, however, how to select the best performing feature set, as simply adding all possible features does not necessarily give the best results (Guyon and Elisseeff, 2003). The approach taken here is to implement a large feature set and then use a greedy search to explore the feature set and select the best subset of features. This method of feature set optimisation is not new (for example, it was applied by one team (Ganchev et al., 2007) on the BioCreative II Gene Mention task), but in this work a comparison of search starting points and feature groupings will be presented.

All RE systems require a human-annotated corpus for testing, and since a supervised machine learning approach is employed, a corpus is also required for training the system. The experiments described in this paper make use of the ITI TXM corpora (Alex et al., 2008), which include the PPI corpus addressing protein-protein interactions, and the TE corpus addressing tissue expression. Both corpora consist of approximately 200 full-text biomedical research papers annotated with entities, normalisations of entities to standard databases, relations, and with enriched information added to the relations. Only the entities and relations will be considered here.

This paper is organised as follows: after reviewing related work in the following section, the RE system is described in Section 3, including a description of the corpora, the relation candidate extraction strategies, the features employed, the feature optimisation methods and the evaluation method. In Section 4 the results of the optimisation experiments are presented and discussed, with some concluding remarks in Section 5.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 19–27,
Columbus, Ohio, USA, June 2008.

2 Related Work

Recent interest in the extraction of protein-protein interactions has been given added impetus by shared tasks such as the Language Learning in Logic (Cussens and Nédellec, 2005), and the BioCreative II Interaction Pairs Subtask (Krallinger et al., 2008). It should be noted that the latter task, rather than being concerned with the extraction of specific interaction relation mentions, required systems to list the (curatable) interactions at a document level. Many teams, however, extracted the interaction mentions as a first step and then processed these to give the document level list of curatable interactions.

The extraction of protein-protein interactions has also been helped by the availability of annotated corpora, such as AIMed (Bunescu et al., 2005), which consists of around 1000 Medline abstracts annotated with proteins and their interactions. In common with the LLL corpus, the AIMed corpus only contains intra-sentential relations, and is somewhat smaller than the corpus used in the current work. In addition to the work by the corpus creators (Bunescu and Mooney, 2007), other authors have achieved good results on AIMed by making use of dependency parses in different ways (Erkan et al., 2007; Katrenko and Adriaans, 2006). It is not clear, however, how well these techniques would transfer to other, similar, RE problems, and how much work would be involved in tuning the systems for a new problem.

Supervised learning based on shallow syntactic features has also been applied to the biomedical domain, again focusing on protein-protein interactions (Nielsen, 2006; Giuliano et al., 2006). A systematic exploration of a set of such features for protein-protein interaction extraction was recently provided by Jiang and Zhai (2007), who also used features derived from the Collins parser. They did not, however, experiment with the automated optimisation of the feature sets. In the news domain, the best reported results on the ACE dataset[1] have been achieved by a composite kernel which depends partially on a full parse, and partially on a collection of shallow syntactic features (Zhou et al., 2007).

Aside from protein-protein interactions, there has been little work directed at other types of relations in the biomedical domain. Recent corpus annotation projects such as Genia (Kim et al., 2008) and BioInfer (Pyysalo et al., 2007) include multiple types of relations, however many of the relation types are represented in fairly small quantities. In earlier work (Skounakis et al., 2003), the extraction of cell localisation relations was studied using an automatically created corpus.

[1] http://www.nist.gov/speech/tests/ace/

3 Methods

3.1 Corpora

The ITI TXM corpora contain annotations related to protein-protein interactions (in the PPI corpus), and annotations related to tissue expression experiments (in the TE corpus). Each corpus consists of biomedical research articles, selected from PubMed and PubMedCentral either because they contain experimentally proven protein-protein interactions (for the PPI corpus), or because they contain tissue expression experiments (for the TE corpus).

The articles were annotated by a team of qualified biologists. The annotations consisted of entities (Table 1), normalisations of selected entities to standard databases, relations (Table 2) and enrichment of relations with additional information of interest to curators. For each corpus, the entities marked were those involved in the relation which formed the principal focus of that corpus (either PPI or TE), and those which could affect this relation. In the TE corpus, TE relations were marked when the text stated that a particular gene or gene product was present or absent in a particular tissue, whilst PPI relations were marked whenever a statement (positive or negative) was made about the interaction of a pair of Proteins, Mutants, Fragments, Complexes or Fusions. In addition, both corpora were annotated with FRAG relations which connect Fragments and Mutants with their parent Proteins.

Corpus	Entities
PPI	CellLine, Complex, DrugCompound, ExperimentalMethod, Fragment, Fusion, Modification, Mutant, Protein
TE	Complex, DevelopmentalStage, Disease, DrugCompound, ExperimentalMethod, Fragment, Fusion, GOMOP, Gene, Mutant, Protein, Tissue, mRNAcDNA

Table 1: The entity types in the TE and PPI corpora. Note that *GOMOP* stands for *"Gene or mRNAcDNA or Protein"* and was used when the annotators felt the author was using the term in an ambiguous way.

In order to monitor annotation quality, and to measure of the difficulty of the task, some documents were multiply annotated. The counts of the numbers of unique documents in each section, together with the numbers of annotated documents are shown in Table 3. Note that the multiply annotated documents were not reconciled, but the multiple copies were included in the corpus. Each corpus was split into three sections – TRAIN, DEVTEST and TEST – with the first two sections being used for system development, and the last reserved for final testing.

Corpus	Relation type	Entity 1 Types	Entity 2 Types	Count
PPI	PPI	Protein, Fusion, Mutant, Fragment or Complex	Protein, Fusion, Mutant, Fragment or Complex	11,523
	FRAG	Protein	Mutant or Fragment	16,002
TE	TE	Gene, Protein, mRNAcDNA, GOMOP, Fusion, Mutant, Complex or Fragment	Tissue	12,426
	FRAG	Protein	Mutant or Fragment	4,735

Table 2: Relation types in each corpus.

Corpus	Segment	Unique Documents	Annotated Documents
PPI	TRAIN	133	221
	DEVTEST	39	58
	TRAIN	45	57
TE	TRAIN	151	221
	DEVTEST	41	48
	TEST	46	59

Table 3: Counts of documents and annotations in each corpus.

Corpus	Relation	Intra	Inter
PPI	PPI	10,607(92.1%)	916(7.9%)
	FRAG	10,176(63.6%)	5,826(36.4%)
TE	TE	10,356(83.3%)	2,070(16.7%)
	FRAG	3,335(70.4%)	1,400(29.6%)

Table 4: Counts of inter and intra-sentential relations.

Annotators were permitted to mark relations between entities in the same sentence (*intra-sentential*), or between entities in different sentences (*inter-sentential*). The majority of relations were intra-sentential, with FRAG relations showing the highest proportion of inter-sententials. Table 4 shows the counts of inter/intra-sentential relations of each type.

Some examples of each type of relation will now be presented. The first example is from PubMed 16436664, and is a TE relation:

> Our recent observations that $\langle \alpha v \beta 5 \rangle_1$ is up-regulated in $\langle$scleroderma fibroblasts$\rangle_1$ and that the transient overexpression of $\alpha v \beta 5$ increases the human $\langle \alpha 2$(I) collagen$\rangle_2$ gene expression in normal $\langle$fibroblasts$\rangle_2$ ...

There are two different TE relations in this sentence fragment, indicated by the numerical subscripts; the first connects a Tissue and a Complex, and the second connects a Tissue with a Gene. Another example from the same paper shows a FRAG relation.

> Because $\langle \beta 5 \rangle_1$ has a $\langle$cytoplasmic domain$\rangle_1$ highly homologous to that of β6-subunit, 42 we made a hypothesis that $\alpha v \beta 5$ activates SLC by the nonproteolytic pathway.

The annotators could also mark negative TE and PPI relations, as shown in the following example of a PPI relation taken from PubMedCentral 1075921.

> It was also previously reported that two truncated versions of $\langle$p53$\rangle_{1,2}$, consisting of residues $\langle$2-45$\rangle_{1,3}$ and $\langle$46-71$\rangle_{2,4}$, do not bind $\langle$hRPA70$\rangle_{3,4}$ (47)

Here the PPI relations connect the two Fragments ("2-45" and "46-71") to the Protein "hRPA70", whilst FRAG relations connect the Fragments with their parent Protein "p53".

In contrast with the straightforward intra-sentential relations shown above, the following (from PubMed 16399077) is an example of an inter-sentential TE relation (only the related entities are shown).

> To test whether SPE can activate Toll signaling, we expressed activated SPE in $\langle$S2 cells$\rangle_1$ and in flies, and we then assayed the expression of the gene for Drosomycin (Drs), an antifungal peptide known to be induced by Toll signaling in response to microbial infection (Lemaitre et al., 1996). In both cases, $\langle$Drs$\rangle_2$ expression was significantly induced in the absence of infection,

In this example, the annotator has connected a Tissue on the first sentence, with an mRNAcDNA in the second.

The multiply annotated documents in the corpus were used to calculate the inter-annotator agreement (IAA), by scoring different versions of the annotation of the same document against each other. For each corresponding pair of annotations, one annotator was selected as the "gold", and the other annotator scored against the first using precision, recall and F_1 on relations. Only relations where both annotators agreed on the participating entities were considered. The scores for each annotated document pair were then micro-averaged (where each example

Corpus	Type	Intra	Inter	All
PPI	PPI	69.7	41.1	67.0
	FRAG	90.5	73.9	84.6
	All	78.7	67.3	76.1
TE	TE	72.8	59.4	70.1
	FRAG	89.7	69.0	84.0
	All	77.4	62.7	74.1

Table 5: IAA for relation annotation, split by inter- and intra-sentential

is given equal weight) to produce overall IAA scores for the corpus, shown in Table 5.

The main observations from Table 5 are that TE and PPI relations are harder to annotate than FRAG relations, and that inter-sentential are harder than intra-sentential. In particular, the IAA for intra-sentential FRAG relations is very high, probably because many of these are very straightforward constructions such as "Fragment of Protein". Inter-sentential relations are often less clear as they involve linking information between several sentences, for example using coreferences.

Both corpora were pre-processed before RE was applied. The pre-processing involved tokenisation, sentence boundary detection, lemmatising. part-of-speech tagging, head word detection and chunking. The part-of-speech tagging uses the Curran & Clark maximum entropy Markov model tagger (Curran and Clark, 2003) trained on MedPost data (Smith et al., 2004), whilst the other preprocessing stages are all rule-based. The tokenisation, sentence boundary detection, head word identification and chunking components were implemented with the LT-XML2 tools (Grover and Tobin, 2006), and the lemmatisation used `morpha` (Minnen et al., 2000).

3.2 The Relation Extraction System

Relation extraction is treated a classification problem, by generating candidate relations, and classifying them as either *true* or *false*. In the optimisation experiments described in this paper, Zhang Le's maximum entropy (MAXENT) classifier[2] was used, since its performance was very competitive and its fast training time permitted extensive feature experimentation. The Gaussian prior was set to 0.1, and the maximum training iterations to 100. In order to assess the performance of the final system, MAXENT was compared with support vector machines (SVM) using the SVM^{light} toolkit (Joachims, 1999). Since both the classifiers assign a confidence to each prediction, a varying threshold can be applied to the output of the classifier to provide a precision-recall tradeoff.

Candidate relations were generated by considering entity pairs of the appropriate type, taking into account the distance between the entities. It was thought that inter-sentential and intra-sentential relations would require different feature sets and different models, so inter- and intra-sentential candidates were generated separately. For intra-sentential relations, all entity pairs of the appropriate type (as in Table 2) in the same sentence were permitted as candidates, with the sole exclusion being that any entities contained in a Fusion entity were not allowed to participate in candidate TE relations. This restriction was in place in the annotation guidelines, so no such relations were annotated. For intra-sentential relations in the training data, around 25-30% of the candidate relations are actual relations.

Generating inter-sentential candidates is more problematic, as measures must be taken to limit the number of candidates. Inter-sentential FRAG candidates are restricted to a distance of no more than 5 sentences, whilst inter-sentential PPI and TE candidates are restricted to participants in adjacent sentences. Inter-sentential RE is performed after intra-sentential RE, so the candidate generation strategy has access to the annotated intra-sentential relations (in training) and the predicted intra-sentential relations (in testing). For TE and PPI, candidates are only created for those entities not already in a relation, and for FRAG candidates are only created if the Mutant or Fragment is not already in a relation. Furthermore, for FRAG relations, if there is more than one Protein instance with the same lexical form in the 5 sentence window, then a candidate relation is only created between a given Fragment/Mutant and the nearest occurrence of this Protein. For inter-sentential FRAG relations, around 20% of the candidates are actual relations, however for TE and PPI, only about 1% of the candidate relations are actual relations.

3.3 Features

Each candidate relation is mapped to a feature representation, where the features are binary or real-valued functions of relations. The majority of the features are binary, although these are actually special cases of real-valued functions, taking values 0 or 1. A feature representation of a relation is normally written as a sequence of strings, each corresponding to a different feature, and the presence or absence of a binary feature indicating whether it is on or off. In order that the relation extractor could be applied to different problems and optimised, a large number of features were implemented, with the intention that the feature space could be automatically searched to find the best subset.

[2] `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`

Features are normally grouped into *feature templates* and, as is common in the literature, the feature templates may also be referred to as *features*. For instance, a feature template may be "the token to the right of the second entity in the relation", which then gives rise to a set of boolean features with the prefix `ctxt-w-rf1-`. One such feature in this set is the feature which indicates that the token to the right of the second entity is "the", i.e. `ctxt-w-rf1-the`. The feature templates are then collected into *feature groups*, such as "context features", which are really just a convenient way of conceptualising, implementing and managing the features, and do not necessarily reflect any common behaviour amongst the features in a group.

The following is a comprehensive list of the feature implemented in the RE system with features listed by group, and the possible options for each feature group given. The options are used to turn on or off feature templates in the group, or change templates, and may be boolean or numerical. The nature of the options will be important in the feature exploration experiments since they influence the type of search operations which may be used to explore the feature space. The features are virtually all domain independent, except for perhaps the SignSlashSign feature which is specific to TE. The RelationKeywordFeature can easily be ported to a new domain by generating a list of keywords appropriate for the given relation.

In the feature group descriptions which follow, the term "participants" refers to the entities within the candidate relation. Some of the features make use of the "vlw backoff", which for a given token is defined as the verb stem, backing off to the lemma if that is not available, backing off to the token itself.

Chunk This group has three optional templates; one which adds the concatenated sequence of chunk types between the participants, and two templates which add the count of chunks between the participants as binary and numeric features, respectively. So if the chunk count is, for example 4, the binary feature would be `chunk-bwcount-4` and the numeric feature would have name `chunk-bwcount` and value 4.

EntitiesBetween This has templates to indicate the type and relative position of the entities between the two participants. For TE relations, only Tissue entities are considered, whilst for other relations only Proteins are considered.

Entity Features derived from the participating entities are added by the templates in this group, which has options to turn on the entity's text, class and bigrams of these. There is also a feature template which adds all words in the entities as separate features, and one that adds all words in the second entity only, plus options to add features which indicate when the two entities have the same textual form, or when one is a substring of the other.

EntityContext The entity context can include tokens, part-of-speech tags, chunk tags and vlw backoffs, each within window sizes determined by numerical options. A further option can switch on a template which adds the concatenation of all vlw backoffs in the context, on either side of each entity, and there is also an option to convert all tokens and vlw backoffs to lower case before creating the features.

EntityDistance Options on this group allow the addition of the token distance and sentence distance between the entities, as numeric or binary features. There is also an option to add a coarse three-way classification of the token distance.

EntityFrequency Counts are made of the number of occurrences of each entity surface form in the document, limited to Tissue entities for TE relations, and Proteins for FRAG and PPI relations. The only option for this feature group adds a template which generates a binary feature indicating the frequency rank of the participants' surface forms in the document.

EntityPattern The entity pattern for a given intrasentential candidate relation shows how its participants lie with respect to the other entities in the sentence. The pattern is a concatenated sequence of the entity types in the sentence, with the participants in upper case and other entities in lower case. Only entity types which are valid participants in the relation in question are included. For example `protein-PROTEIN-TISSUE` would indicate a relation between a Protein and a Tissue, with another Protein occurring first in the sentence. Options in this group add the patterm, the total number of entities in the sentence, and the numbers of entities for each type.

Frame The frame is the concatenation of the tokens between the two participants. Two boolean options on this group specify whether or not to include the token concatenation, and whether or not to include the part-of-speech concatenation. A further numeric option is used to limit the maximum frame length; when this is set to a non-zero value longer frames are discarded.

HeadWord All the headwords of the chunks in the sentences containing and between the participants are listed and used to construct the features in this group. Options specify whether to include head nouns and/or head verbs, and whether to convert the headwords to lower case or replace them by their vlw backoffs. A further option allows an additional marker to be added to each headword feature to indicate whether it is before, between or after the participants.

NestedEntity This feature indicates whether the participants are contained in other entities, or in each

other. The first option adds a feature template which indicates which type of entity containing the two participants, if they are both contained. The second option adds a feature to indicate whether one of the entities is contained within the other, and the third adds a feature to indicate whether or not there is any whitespace between the two entities.

Ngram Three options specify what type of ngrams to add; whether to add unigrams of the tokens in the sentences containing the participants, whether to add bigrams of the same tokens, and whether to add cross-bigrams, which are bigrams of tokens before and between the participants, and of tokens between and after the participants. Additional options specify whether to convert tokens to vlw backoff or lower case and whether to replace all sequences of digits by "0". Further options can be used to indicate that only ngrams in between the participants should be added, that each ngram feature should be marked as before, between or after, or that all entities should replaced in the text by their type.

RelationKeyword Relation keywords are terms annotated as relation indicators for PPI and TE, and linked to relations. For PPI they are interaction words, and for TE they are expression level words. Keywords are matched from a list generated during training and there are feature options to match these keywords before, between and after the participants, and to add templates for the existence of a keyword, the text of the keyword, and whether or not it is a head word.

RelativeEntityPosition The only option on this group specifies whether or not to sort the participant entities, alphabetically by entity type. Binary features are added indicating whether the first entity in the candidate relation is the first in the document, whether it is the second, whether the participants overlap or whether they coincide.

SignSlashSign This group is only used for TE relations and is designed to detected the presence of indicators like +/+ and −/+ in the sentence(s) containing the relation. Options allow the existence and type of the one of these expressions to be indicated, and also its position relative to the participants, and whether it is adjacent to one of the participants.

3.4 Optimisation

Feature selection methods include *wrapper* methods where feature sets are assessed according to their effectiveness for a given learner, and *filter* methods where features are removed using some criterion before being passed to the learner (Guyon and Elisseeff, 2003). In building the RE system, it was found that filter methods did not work well, probably due to the large number of interactions between the features, so a wrapper optimisation method was employed, consisting of greedy search through the space of possible feature sets.

In the greedy search method, an initial feature set is selected and a model trained on the TRAIN set and tested on the DEVTEST set. A series of search operators (see below) are applied to the feature set to produce a list of proposed new feature sets, one corresponding to each operator, and the new feature sets are tested in the same way. If any of these new feature sets produces better results than the original initial set, then the best set replaces the initial feature set and the process is iterated. The greedy search terminates when none of the search operators leads to an improvement. Three types of search operators are used in the greedy search, defined in terms of the feature set structure described in Section 3.3:

1. The deletion of a feature group.
2. The increase or decrease of a numerical option on a feature group (e.g. context size), where the size of the change is not greater than 2.
3. The flipping of a boolean option on a feature group.

In theory search operators which add or remove individual features could be used, but due to the large number of features the use of such operators is not practical. In addition, it may have been possible to achieve more robust results using cross-validation rather than heldout testing, but that would also result in a large increase in search time.

3.5 Evaluation

In all RE experiments, the annotated entities were assumed as given so that only RE performance was being assessed. The performance was measured using *precision-recall break-even point* (BEP), which is found by adjusting the decision boundary (threshold) of the classifier until the precision and recall are equal then taking the value of the F_1 at this threshold. The BEP has the advantage over F_1 that its definition is independent of the choice of threshold, but it can still be compared easily to the IAA and is based on the familiar concepts of precision and recall.

4 Results

Performance of the RE system on each of the four relation types was optimised using the greedy feature exploration method described in Section 3.4. Inter and intra-sentential relations were treated separately, with intra-sentential relation performance optimised first. The inter-sentential performance was then assessed using a "pipeline" consisting of the best intra-sentential relation extractor, and the inter-sentential system being optimised.

The greedy search experiments for intra-sentential

relations used two different starting feature sets, an **all** set in which all features groups and options were switched on, and the context sizes in *EntityContext* were set to 3, and a **base** set which used just *Ngram* and *RelativeEntityPosition* features. The models were trained on TRAIN and scored on DEVTEST using BEP. In the calculation of BEP, all relations of the appropriate type were considered, including inter-sententials. The results of the greedy search on intra-sentential relations are shown in Table 6.

Corpus	Relation Type	Initial Features	Initial BEP	Final BEP
PPI	PPI	**base**	36.8	52.2
		all	51.6	53.4
	FRAG	**base**	49.2	56.0
		all	55.9	57.4
TE	TE	**base**	45.9	51.9
		all	50.6	53.8
	FRAG	**base**	53.7	62.7
		all	60.1	61.2

Table 6: Greedy search feature exploration for intra-sentential relations. Performance is measured on all relations, testing on DEVTEST.

For all relation types, the greedy search improves the performance over the **base** and **all** feature sets, usually reaching the highest performance when starting from **all**. Comparing the results in Table 6 with the IAA figures provided in Table 5 shows that the system performance is around 75-80% of IAA, with the lowest relative performances observed for FRAG relations. These relations include a higher proportion of inter-sententials, so systems which ignore inter-sententials suffer a larger loss in performance.

After choosing the best system for intra-sentential relations, the same greedy optimisation was performed on the inter-sentential relations using virtually the same initial feature sets. The only difference in the feature sets is that additional options are added to the *EntityDistance* feature to indicate the sentential distance between the entities. The result of the greedy search on the inter-sentential relations is shown in Table 7.

The inter-sentential relation optimisation is only really successful for the FRAG relations in the PPI corpus. For TE and PPI inter-sentential relations, the number of negative examples dwarfs the few positive examples making it very difficult for the machine learner. For FRAG relations in both corpora, some progress is made on the performance on inter-sentential relations (detailed breakdown not shown) but in the TE corpus this does not translate to an overall improvement in BEP. This is because the inter- and intra-sentential probabilities have quite

Corpus	Relation Type	Initial Features	Initial BEP	Final BEP
PPI	PPI	**base**	53.4	53.4
		all	53.4	53.4
	FRAG	**base**	59.6	62.2
		all	61.7	62.5
TE	TE	**base**	53.9	54.0
		all	53.9	54.0
	FRAG	**base**	60.4	62.8
		all	62.6	62.7

Table 7: Greedy search feature exploration for inter-sentential relations. Performance is measured on all relations, testing on DEVTEST.

different ranges for FRAG relations meaning that the threshold probabilities would have to be chosen separately to give the best F_1 score.

The greedy search results just presented were based on a partitioning of the feature sets into groups which correspond to the way in which the features were implemented. Since the search operators apply at group granularity, and are not able to select features from within a group, the way in which the features are grouped is likely to have a bearing on the performance of the best system found by the algorithm. The next set of experiments investigates the effective the feature grouping by conducting greedy search with groups chosen randomly.

Corpus	Relation Type	Initial BEP	Final BEP	Ensemble BEP
PPI	PPI	51.1	52.9, 52.4, 52.7, 52.8, 52.6	52.5
	FRAG	55.7	56.3, 56.1, 56.1, 56.3, 56.4	56.3
TE	TE	51.4	52.0 , 51.8, 52.5, 51.9, 52.9	52.1
	FRAG	60.1	60.8, 60.5, 60.4, 60.7, 60.5	60.4

Table 8: Greedy search feature exploration with random feature groupings for intra-sentential relations. The initial feature set is a slightly modified **all** in each case, and the search was run 5 times, testing on DEVTEST. The ensemble system combines the 5 optimised feature sets using the geometric mean probability.

Using a variant of the **all** feature set where the context sizes in *EntityContext* were set to 5, a greedy search for the best performing system was implemented by first dividing the feature set randomly into 50 groups, and at each iteration testing the performance with each group added and removed in turn. The search was iterated until no further improvement in performance was obtained, where

performance was measured using BEP. As for the previous greedy feature optimisations, the relation extractor was trained on TRAIN and tested on DEVTEST. The results for intra-sentential relations are shown in Table 8, where the experiment was repeated several times with different (randomly chosen) partitions. After performing the five random knockout searches of the feature space, an ensemble system was created for each relation type by training a system with each feature set and combining the five by taking the geometric mean of the probabilities. The performance of the ensemble system is shown in the final column of Table 8.

Comparing the results in Table 8 with the corresponding results for intra-sentential relations in Table 6, it can be seen that splitting the features into related groups works better than random groups. The ensemble does not improve on the individual scores, probably because the systems in the ensemble are not diverse enough (Dietterich, 2000)

To see how well the best feature sets generalise to unseen data, RE systems were trained on TRAIN and DEVTEST combined, and tested on TEST using different feature sets; the baseline sets (**base** and **all**), and the fully optimised set (**best**). In addition, to ensure that the greedy feature optimisation was not biasing the feature set towards the particular learner employed (i.e. MAXENT), systems were also trained and tested using SVM. The MAXENT system had its Gaussian prior optimised on the DEVTEST set, whilst SVM was found to work best with a linear kernel, and its cost factor was optimised on DEVTEST. The value of the decision function was used for thresholding the SVM model in order to calculate the BEP. The comparison of all systems on TEST is shown in Table 9.

Corpus	Relation Type	Learner	Feature Set		
			base	**all**	**best**
PPI	PPI	MAXENT	39.7	48.3	49.1
		SVM	39.6	49.2	**49.9**
PPI	FRAG	MAXENT	56.9	68.0	69.4
		SVM	54.9	68.2	**69.5**
TE	TE	MAXENT	39.0	47.9	46.8
		SVM	39.6	49.8	**50.1**
TE	FRAG	MAXENT	60.1	63.4	68.9
		SVM	59.7	67.7	**70.4**

Table 9: The performance of the system trained on TRAIN and DEVTEST, and tested on TEST. Performance is compared across the baseline feature sets (**base** and **all**) and the optimised feature set (**best**) using each classifier.

The results in Table 9 show that, in general, both classifiers perform better with the **all** feature set than with the **base** feature set, and best of all with the **best** feature set. The SVM classifier preserves this ordering throughout, and actually performs better than the MAXENT classifier overall, even though the features were optimised for MAXENT. For MAXENT, the **best** model outperforms **all** in three out of four cases, with the exception being TE.

5 Conclusions

It has been shown that a relation extraction system based on a supervised classifier and a large collection of shallow linguistic features can be applied to three different types of relations in two different biomedical corpora. Automated feature optimisation produced small gains in performance which were still apparent on a blind test set. Even though a wrapper method was used using a specific classifier (MAXENT), the feature set optimisations were still valid for an SVM classifier.

Since the greedy search through feature space is essentially a beam search with a beam size of one, it could be extended by using a larger beam-size, running the feature set comparisons in parallel to reduce total running time to a manageable size. Ad-hoc experiments have suggested that better results could be obtained by restarting the feature optimisation in different positions, indicating that local optima could be a problem, but a thorough investigation of the search space nature has been left for future work. Furthermore, the hyperparameter optimisation of the classifiers (for example the Gaussian prior in MAXENT) could be incorporated into the search.

Whilst the relation extractor was successful on intra-sentential relations, it is less successful on inter-sentential relations, perhaps becuase of the lingusitic complexity of these, and the sparsity of positive examples. The split into inter- and inter-sentential examples in the current system seems justified as they have quite different characteristic, but there may also be a case for splitting the intra-sententials further, into intra- and inter-clausals, as suggested by Maslennikov and Chua (2007), and then treating inter-clausals and inter-sententials together. Whilst intra-clausals are more likely to use simple constructions and be amenable to modelling with shallow linguistic features, inter-sententials and inter-clausals are more likely to use complex linguistic phenomena such as corefereces.

Acknowledgements

This work was supported by the Text Mining Programme of ITI Life Sciences Scotland (`http://www.itilifesciences.com`).

References

Bea Alex, Claire Grover, Barry Haddow, Mijail Kabadjov, Ewan Klein, Michael Matthews, Stuart Roebuck, Richard Tobin, and Xinglong Wang. 2008. The ITI TXM Corpora: Tissue Expressions and Protein-Protein Interactions. In *Proceedings of LREC*.

Razvan C. Bunescu and Raymond J. Mooney. 2007. Extracting relations from text: From word sequences to dependency paths. In Anne Kao and Steve Poteet, editors, *Text Mining and Natural Language Processing*, pages 29–44. Springer.

Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani, and Yuk W. Wong. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.

James Curran and Stephen Clark. 2003. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL*.

James Cussens and Claire Nédellec, editors. 2005. *Proceedings of Language Learning in Logic*.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15.

Gunes Erkan, Arzucan Ozgur, and Dragomir R. Radev. 2007. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In *Proceedings of EMNLP-CoNLL*.

Kuzman Ganchev, Koby Crammer, Fernando Pereira, Gideon Mann, Kedar Bellare, Andrew McCallum, Steven Carroll, Yang Jin, and Peter White. 2007. Penn/UMass/CHOP Biocreative II systems. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*.

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of EACL*.

Claire Grover and Richard Tobin. 2006. Rule-based chunking and reusability. In *Proceedings of LREC*.

Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182.

Jing Jiang and Chengxiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Proceedings of NAACL*.

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. In *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA.

S. Katrenko and P. W. Adriaans. 2006. Learning relations from biomedical corpora using dependency tree levels. In *Proceedings of Benelearn*.

Jin D. Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1).

Martin Krallinger, Florian Leitner, Carlos Rodriguez-Penagos, and Alfonso Valencia. 2008. Overview of the protein-protein interaction annotation extraction task of BioCreative II. *Genome Biology (in press)*.

Mstislav Maslennikov and Tat S. Chua. 2007. A multi-resolution framework for information extraction from free text. In *Proceedings of ACL*.

Guido Minnen, John Carroll, and Darren Pearce. 2000. Robust, applied morphological generation. In *Proceedings of INLG*.

Leif Arda Nielsen. 2006. Extracting protein-protein interactions using simple contextual features. In *Proceedings of BioNLP*.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Bjorne, Jorma Boberg, Jouni Jarvinen, and Tapio Salakoski. 2007. Bioinfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(1).

Marios Skounakis, Mark Craven, and Soumya Ray. 2003. Hierarchical hidden markov models for information extraction. In Georg Gottlob, Toby Walsh, Georg Gottlob, and Toby Walsh, editors, *Proceedings of IJCAI*.

L. Smith, T. Rindflesch, and W. J. Wilbur. 2004. MedPost: a part-of-speech tagger for biomedical text. *Bioinformatics*, 20(14):2320–2321.

Guodong Zhou, Min Zhang, Donghong Ji, and Qiaoming Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of EMNLP-CoNLL*.

Mining the Biomedical Literature for Genic Information

Catalina O. Tudor ⋆ **K. Vijay-Shanker** ⋆ **Carl J. Schmidt** ◦
Department of Computer and Information Sciences ⋆
Department of Animal and Food Sciences ◦
University of Delaware, Newark, DE 19716
{tudor,vijay}@cis.udel.edu schmidtc@udel.edu

Abstract

eGIFT (**E**xtracting **G**ene **I**nformation **F**rom **T**ext) is an intelligent system which is intended to aid scientists in surveying literature relevant to genes of interest. From a gene specific set of abstracts retrieved from PubMed, eGIFT determines the most important terms associated with the given gene. Annotators using eGIFT can quickly find articles describing gene functions and individuals scientists surveying the results of high-throughput experiments can quickly extract information important to their hits.

1 Introduction

Given the huge number of articles from the biomedical domain, it has become very difficult for scientists to quickly search and find the information they need. Systems to facilitate literature search are being built. E.g. GoPubMed (Doms and Schroeder, 2005) clusters abstracts retrieved from PubMed based on GO and MeSH terms, iHOP (Hoffman and Valencia, 2005) connects biomedical literature based on genes, EBIMed (Rebholz-Schuhmann et al., 2006) displays sentences containing GO terms, drugs, and species.

In contrast to these systems, eGIFT automatically identifies the most relevant terms associated with a given gene. We believe that such a retrieval of terms could itself enable the scientists to form a reasonable good idea about the gene. For example, some of the top key phrases associated with *Groucho* (Entrez Gene ID 43162) by eGIFT are: *transcriptional corepressor*, *segmentation*, *neurogenesis* and *wd40*. This might immediately inform a user that *Groucho* is probably a *transcriptional corepressor*, that it might be involved in the processes of *segmentation* and *neurogenesis* and that it might contain the *wd40* domain, which allows them to draw further inferences about the gene. To enable the scientists to get a deeper understanding, eGIFT further allows the retrieval of all sentences from this gene's literature containing the key phrase in question. The sentences can be displayed in isolation or in the context of the abstract in which they appear.

2 Ranking Key Terms

(Andrade and Valencia, 1998) automatically extracted keywords from scientific text by computing scores for each word in a given protein family, based on the frequency of the word in the family, the average frequency of the word and the deviation of word distribution over all families. (Liu et al., 2004) extended this method to statistically mine functional keywords associated with genes.

Our application is somewhat similar in that we compare the distribution of phrases in the abstracts about the gene from some background set. We use statistical methods to identify the situations where the different frequencies of appearance of a term in two sets of the literature are statistically interesting. We differ from the above work by choosing a broader range of background information. Our motivation is to retrieve any type of phrases, thus not limiting ourselves to only functional terms or terms that might differentiate the selected set of protein families. Since we no longer have several sets of litera-

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 28–29,
Columbus, Ohio, USA, June 2008.

ture, our approach differs from the above method in that we cannot base the score on average frequencies and term deviation in the same way.

Background Set (BSet): In order to capture a wide range of information about genes in general, we downloaded from PubMed all the abstracts for the following boolean query: gene[tiab] OR genes[tiab] OR protein[tiab] OR proteins[tiab]. Approximately 640,000 non-empty abstracts were found.

Query Set (QSet): We download from PubMed the abstracts that mention a given gene name and its synonyms. We obtained the latter from BioThesaurus (Liu et al., 2005).

Key Term Scores: We considered many different statistical tests to identify significant key phrases, but eventually settled on the following score:

$$s_t = (\frac{dc_{tq}}{N_q} - \frac{dc_{tb}}{N_b}) * ln\left(\frac{N_b}{dc_{tb}}\right)$$

where dc_{tb} and dc_{tq} are the background and query document counts of term t, and N_b and N_q are the total number of documents from the BSet and QSet.

The difference in frequencies ($\frac{dc_{tq}}{N_q} - \frac{dc_{tb}}{N_b}$) gives preference to terms that appear more frequently in the QSet than in the BSet. This way, we would like to capture terms that are common to the given gene but not to genes and proteins in general. The difference itself is not sufficient to eliminate common words. To address this problem, similar to the use of IDF in IR, we add a global frequency term ($ln\left(\frac{N_b}{dc_{tb}}\right)$) to further penalize common terms, such as *protein*.

To better understand how the score is computed, consider the gene *Groucho* and its key term *corepressor*, which was mentioned in 66% of the QSet and only in 0.1% of the BSet. The huge difference in frequencies, together with the low background frequency, helped the key term *corepressor* score 4.3617, while most of the terms score below 0.25.

Enhancements to Basic Method: First, we extended our method to include unigrams, bigrams, and multi-word terms where previously identified. We observed that some words are not meaningful when presented alone. For instance, the words *development* and *embryonic* taken separately are not as informative as when put together into *embryonic development*, a term which was ranked much higher than the two words.

Next, we applied morphological grouping on terms, based on manually developed rules, after observing variances within the same concept. In writing, we can say *corepressor*, *co-repressor*, or *co-repressors*. In order to capture the concept, we computed frequencies on morphological groups and not on each individual term.

Last, we divided key terms into categories by using morphological information to separate terms such as descriptors, and by consulting publicly available controlled vocabularies (such as NCBI Conserved Domains, NCBI Taxonomy, MedlinePlus, DrugBank, and MeSH category A01).

3 Assessment

Our method has been applied on 55 different genes selected by annotators for a public resource. The initial feedback has been encouraging. Also preliminary investigations suggest we get far more keywords associated with some genes in resources such as GenBank, SwissProt and Gene Ontology than the system of (Liu et al., 2004). Our next goal is to do a thorough evaluation of our system.

References

Miguel A Andrade and Alfonso Valencia. 1998. Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. *Bioinformatics*, 14(7):600–607.

Andreas Doms and Michael Schroeder. 2005. GoPubMed: exploring PubMed with the Gene Ontology. *Nucleid Acid Research*, 33:w783–w786.

Robert Hoffman and Alfonso Valencia. 2005. Implementing the iHOP concept for navigation of biomedical literature. *Bioinformatics*, 21:ii252–ii258.

Ying Liu, Martin Brandon, Shamkant Navathe, Ray Dingledine, and Brian J. Ciliax. 2004. Text mining functional keywords associated with genes. *MedInfo*, 11:292–296.

Hongfang Liu, Zhang-Zhiu Hu, Jian Zhang, and Cathy Wu. 2005. Biothesaurus: a web-based thesaurus of protein and gene names. *Bioinformatics*, 22(1):103–105.

Dietrich Rebholz-Schuhmann, Harald Kirsch, Miguel Arregui, Sylvain Gaudan, Mark Riethoven, and Peter Stoehr. 2006. EBIMed - text crunching to gather facts for proteins from Medline. *Bioinformatics*, 23:e237–e244.

Accelerating the Annotation of Sparse Named Entities by Dynamic Sentence Selection

Yoshimasa Tsuruoka[1], **Jun'ichi Tsujii**[1,2,3] **and Sophia Ananiadou**[1,3]
[1] School of Computer Science, The University of Manchester, UK
[2] Department of Computer Science, The University of Tokyo, Japan
[3] National Centre for Text Mining (NaCTeM), Manchester, UK
yoshimasa.tsuruoka@manchester.ac.uk
tsujii@is.s.u-tokyo.ac.jp
sophia.ananiadou@manchester.ac.uk

Abstract

This paper presents an active learning-like framework for reducing the human effort for making named entity annotations in a corpus. In this framework, the annotation work is performed as an iterative and interactive process between the human annotator and a probabilistic named entity tagger. At each iteration, sentences that are most likely to contain named entities of the target category are selected by the probabilistic tagger and presented to the annotator. This iterative annotation process is repeated until the estimated coverage reaches the desired level. Unlike active learning approaches, our framework produces a named entity corpus that is free from the sampling bias introduced by the active strategy. We evaluated our framework by simulating the annotation process using two named entity corpora and show that our approach could drastically reduce the number of sentences to be annotated when applied to sparse named entities.

1 Introduction

Named entities play a central role in conveying important domain specific information in text, and good named entity recognizers are often required in building practical information extraction systems. Previous studies have shown that automatic named entity recognition can be performed with a reasonable level of accuracy by using various machine learning models such as support vector machines (SVMs) or conditional random fields (CRFs) (Tjong Kim Sang and De Meulder, 2003; Settles, 2004; Okanohara et al., 2006).

However, the lack of annotated corpora, which are indispensable for training machine learning models, makes it difficult to broaden the scope of text mining applications. In the biomedical domain, for example, several annotated corpora such as GENIA (Kim et al., 2003), PennBioIE (Kulick et al., 2004), and GENETAG (Tanabe et al., 2005) have been created and made publicly available, but the named entity categories annotated in these corpora are tailored to their specific needs and not always sufficient or suitable for text mining tasks that other researchers need to address.

Active learning is a framework which can be used for reducing the amount of human effort required to create a training corpus (Dagan and Engelson, 1995; Engelson and Dagan, 1996; Thompson et al., 1999; Shen et al., 2004). In active learning, samples that need to be annotated by the human annotator are picked up by a machine learning model in an iterative and interactive manner, considering the informativeness of the samples. Active learning has been shown to be effective in several natural language processing tasks including named entity recognition.

The problem with active learning is, however, that the resulting annotated data is highly dependent on the machine learning algorithm and the sampling strategy employed, because active learning annotates only a *subset* of the given corpus. This sampling bias is not a serious problem if one is to use the annotated corpus only for their own machine learning purpose and with the same machine learning algorithm. However, the existence of bias is not desirable if one also wants the corpus to be used by other applications or researchers. For the same reason, ac-

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 30–37,
Columbus, Ohio, USA, June 2008.

tive learning approaches cannot be used to enrich an existing linguistic corpus with a new named entity category.

In this paper, we present a framework that enables one to make named entity annotations for a given corpus with a reduced cost. Unlike active learning approaches, our framework aims to annotate *all* named entities of the target category contained in the corpus. Obviously, if we were to ensure 100% coverage of annotation, there is no way of reducing the annotation cost, i.e. the human annotator has to go through every sentence in the corpus. However, we show in this paper that it is possible to reduce the cost by slightly relaxing the requirement for the coverage, and the reduction can be drastic when the target named entities are sparse.

We should note here that the purpose of this paper is not to claim that our approach is superior to existing active learning approaches. The goals are different—while active learning aims at optimizing the performance of the resulting machine learning-based tagger, our framework aims to help develop an unbiased named entity-annotated corpus.

This paper is organized as follows. Section 2 describes the overall annotation flow in our framework. Section 3 presents how to select sentences using the output of a probabilistic tagger. Section 4 describes how to estimate the coverage during the course of annotation. Experimental results using two named entity corpora are presented in section 5. Section 6 describes related work and discussions. Concluding remarks are given in section 7.

2 Annotating Named Entities by Dynamic Sentence Selection

Figure 1 shows the overall flow of our annotation framework. The framework is an iterative process between the human annotator and a named entity tagger based on CRFs. In each iteration, the CRF tagger is trained using all annotated sentences available and is applied to the unannotated sentences to select sentences that are likely to contain named entities of the target category. The selected sentences are then annotated by the human annotator and moved to the pool of annotated sentences.

This overall flow of annotation framework is very similar to that of active learning. In fact, the only differences are the criterion of sentence selection and the fact that our framework uses the estimated coverage as the stopping condition. In active learning, sentences are selected according to their informativeness to the machine learning algorithm. Our approach, in contrast, selects sentences that are most likely to contain named entities of the target category. Section 3 elaborates on how to select sentences using the output of the CRF-based tagger.

1. Select the first n sentences from the corpus and annotate the named entities of the target category.
2. Train a CRF tagger using all annotated sentences.
3. Apply the CRF tagger to the unannotated sentences in the corpus and select the top n sentences that are most likely to contain target named entities.
4. Annotate the selected sentences.
5. Go back to 2 (repeat until the estimated coverage reaches a satisfactory level).

Figure 1: Annotating named entities by dynamic sentence selection.

The other key in this annotation framework is when to stop the annotation work. If we repeat the process until all sentences are annotated, then obviously there is not merit of using this approach. We show in section 4 that we can quite accurately estimate how much of the entities in the corpus are already annotated and use this estimated coverage as the stopping condition.

3 Selecting Sentences using the CRF tagger

Our annotation framework takes advantage of the ability of CRFs to output multiple probabilistic hypotheses. This section describes how we obtain named entity candidates and their probabilities from CRFs in order to compute the expected number of named entities contained in a sentence [1].

[1]We could use other machine learning algorithms for this purpose as long as they can produce probabilistic output. For

3.1 The CRF tagger

CRFs (Lafferty et al., 2001) can be used for named entity recognition by representing the spans of named entities using the "BIO" tagging scheme, in which 'B' represents the beginning of a named entity, 'I' the inside, and 'O' the outside (See Table 2 for example). This representation converts the task of named entity recognition into a sequence tagging task.

A linear chain CRF defines a single log-linear probabilistic distribution over the possible tag sequences $\mathbf{y}$ for a sentence $\mathbf{x}$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(t, y_t, y_{t-1}, \mathbf{x_t}),$$

where $f_k(t, y_t, y_{t-1}, \mathbf{x_t})$ is typically a binary function indicating the presence of feature k, λ_k is the weight of the feature, and $Z(X)$ is a normalization function:

$$Z(\mathbf{x}) = \sum_{y} \exp \sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(t, y_t, y_{t-1}, \mathbf{x_t}).$$

This modeling allows us to define features on states ("BIO" tags) and edges (pairs of adjacent "BIO" tags) combined with observations (e.g. words and part-of-speech (POS) tags).

The weights of the features are determined in such a way that they maximize the conditional log-likelihood of the training data[2] $\mathcal{L}(\theta) = \sum_{i=1}^{N} \log p_\theta(\mathbf{y}^{(i)}|\mathbf{x}^{(i)})$. We use the L-BFGS algorithm (Nocedal, 1980) to compute those parameters.

Table 1 lists the feature templates used in the CRF tagger. We used unigrams of words/POS tags, and prefixes and suffixes of the current word. The current word is also normalized by lowering capital letters and converting all numerals into '#', and used as a feature. We created a word shape feature from the current word by converting consecutive capital letters into 'A', small letters 'a', and numerals '#'.

example, maximum entropy Markov models are a possible alternative. We chose the CRF model because it has been proved to deliver state-of-the-art performance for named entity recognition tasks by previous studies.

[2]In the actual implementation, we used L2 norm penalty for regularization.

Word Unigram	w_i, w_{i-1}, w_{i+1}	& y_i
POS Unigram	p_i, p_{i-1}, p_{i+1}	& y_i
Prefix, Suffix	prefixes of w_i	& y_i
	suffixes of w_i	& y_i
	(up to length 3)	
Normalized Word	N(w_i)	& y_i
Word Shape	S(w_i)	& y_i
Tag Bi-gram	true	& $y_{i-1}y_i$

Table 1: Feature templates used in the CRF tagger.

3.2 Computing the expected number of named entities

To select sentences that are most likely to contain named entities of the target category, we need to obtain the *expected number* of named entities contained in each sentence. CRFs are well-suited for this task as the output is fully probabilistic.

Suppose, for example, that the sentence is "Transcription factor GATA-1 and the estrogen receptor". Table 2 shows an example of the 5-best sequences output by the CRF tagger. The sequences are represented by the aforementioned "BIO" representation. For example, the first sequence indicates that there is one named entity 'Transcription factor' in the sequence. By summing up these probabilistic sequences, we can compute the probabilities for possible named entities in a sentence. From the five sequences in Table 2, we obtain the following three named entities and their corresponding probabilities.

'Transcription factor' (0.677 + 0.242 = 0.916)
'estrogen receptor' (0.242 + 0.009 = 0.251)
'Transcription factor GATA-1' (0.012 + 0.009 = 0.021)

The expected number of named entities in this sentence can then be calculated as 0.916 + 0.251 + 0.021 = 1.188.

In this example, we used 5-best sequences as an approximation of all possible sequences output by the tagger, which are needed to compute the exact expected number of entities. One possible way to achieve a good approximation is to use a large N for N-best sequences, but there is a simpler and more efficient way [3], which directly produces the exact

[3]We thank an anonymous reviewer for pointing this out.

Probability	Transcription	factor	GATA-1	and	the	estrogen	receptor
0.677	B	I	O	O	O	O	O
0.242	B	I	O	O	O	B	I
0.035	O	O	O	O	O	O	O
0.012	B	I	I	O	O	O	O
0.009	B	I	I	O	O	B	I
:	:	:	:	:	:	:	:

Table 2: N-best sequences output by the CRF tagger.

expected number of entities. Recall that named entities are represented with the "BIO" tags. Since one entity always contains one "B" tag, we can compute the number of expected entities by simply summing up the marginal probabilities for the "B" tag on each token in the sentence[4].

Once we compute the expected number of entities for every unannotated sentence in the corpus, we sort the sentences in descending order of the expected number of entities and choose the top n sentences to be presented to the human annotator.

4 Coverage Estimation

To ensure the quality of the resulting annotated corpus, it is crucial to be able to know the current coverage of annotation at each iteration in the annotation process. To compute the coverage, however, one needs to know the total number of target named entities in the corpus. The problem is that it is not known until all sentences are annotated.

In this paper, we solve this dilemma by using an estimated value for the total number of entities. Then, the estimated coverage can be computed as follows:

$$(estimated\_coverage) = \frac{m}{m + \sum_{i \in U} E_i} \quad (1)$$

where m is the number of entities actually annotated so far and E_i is the expected number of entities in sentence i, and U is the set of unannotated sentences in the corpus. At any iteration, m is always known and E_i is obtained from the output of the CRF tagger as explained in the previous section.

[4]The marginal probabilities on each token can be computed by the forward-backward algorithm, which is much more efficient than computing N-best sequences for a large N.

	# Entities	Sentences (%)
CoNLL: LOC	7,140	5,127 (36.5%)
CoNLL: MISC	3,438	2,698 (19.2%)
CoNLL: ORG	6,321	4,587 (32.7%)
CoNLL: PER	6,600	4,373 (31.1%)
GENIA: DNA	2,017	5,251 (28.3%)
GENIA: RNA	225	810 (4.4%)
GENIA: cell_line	835	2,880 (15.5%)
GENIA: cell_type	1,104	5,212 (28.1%)
GENIA: protein	5,272	13,040 (70.3%)

Table 3: Statistics of named entities.

5 Experiments

We carried out experiments to see how our method can improve the efficiency of annotation process for sparse named entities. We evaluate our method by simulating the annotation process using existing named entity corpora. In other words, we use the gold-standard annotations in the corpus as the annotations that would be made by the human annotator during the annotation process.

5.1 Corpus

We used two named entity corpora for the experiments. One is the training data provided for the CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003), which consists of 14,041 sentences and includes four named entity categories (LOC, MISC, ORG, and PER) for the general domain. The other is the training data provided for the NLPBA shared task (Kim et al., 2004), which consists of 18,546 sentences and five named entity categories (DNA, RNA, cell_line, cell_type, and protein) for the biomedical domain. This corpus is created from the GENIA corpus (Kim et al., 2003) by merging the original fine-grained named entity categories.

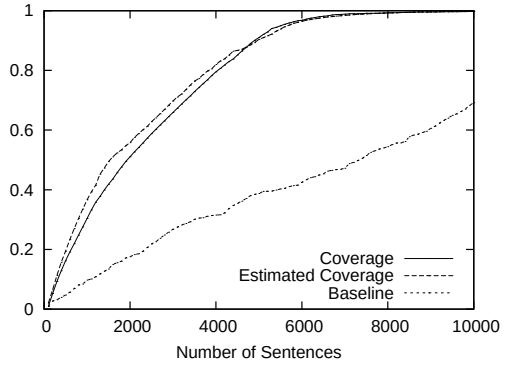

Figure 2: Annotation of LOC in the CoNLL corpus.

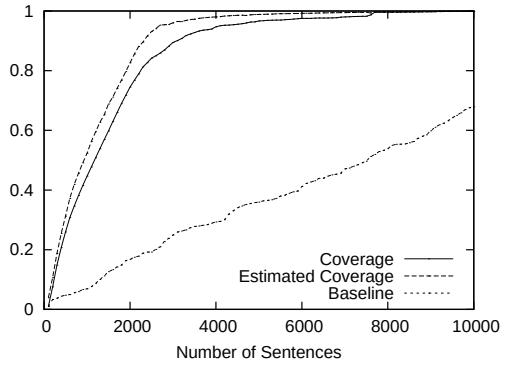

Figure 3: Annotation of MISC in the CoNLL corpus.

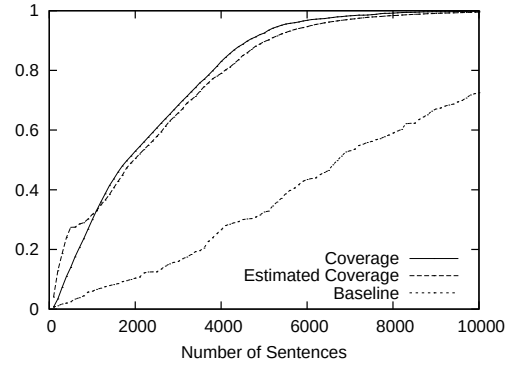

Figure 4: Annotation of ORG in the CoNLL corpus.

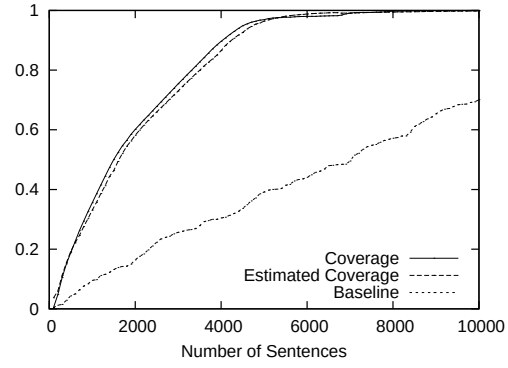

Figure 5: Annotation of PER in the CoNLL corpus.

Table 3 shows statistics of the named entities included in the corpora. The first column shows the number of named entities for each category. The second column shows the number of the sentences that contain the named entities of each category. We can see that some of the named entity categories are very sparse. For example, named entities of "RNA" appear only in 4.4% of the sentences in the corpus. In contrast, named entities of "protein" appear in more than 70% of the sentences in the corpus.

In the experiments reported in the following sections, we do not use the "protein" category because there is no merit of using our framework when most sentences are relevant to the target category.

5.2 Results

We carried out eight sets of experiments, each of which corresponds to one of those named entity categories shown in Table 3 (excluding the "protein" category). The number of sentences selected in each iteration (the value of n in Figure 1) was set to 100 throughout all experiments.

Figures 2 to 5 show the results obtained on the CoNLL data. The figures show how the coverage increases as the annotation process proceeds. The x-axis shows the number of annotated sentences.

Each figure contains three lines. The normal line represents the coverage actually achieved, which is computed as follows:

$$(coverage) = \frac{entities\_annotated}{total\_number\_of\_entities}. \quad (2)$$

The dashed line represents the coverage estimated by using equation 1. For the purpose of comparison, the dotted line shows the coverage achieved by the baseline annotation strategy in which sentences are selected sequentially from the beginning to the end in the corpus.

The figures clearly show that our method can drastically accelerate the annotation process in comparison to the baseline annotation strategy. The improvement is most evident in Figure 3, in which

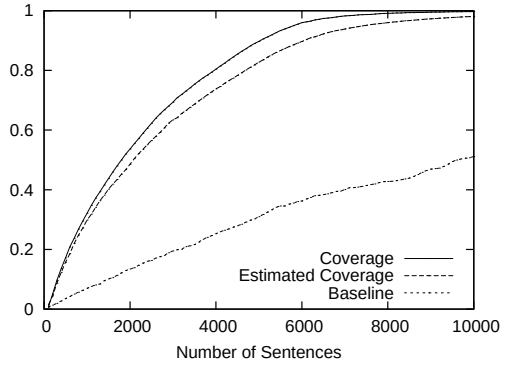

Figure 6: Annotation of DNA in the GENIA corpus.

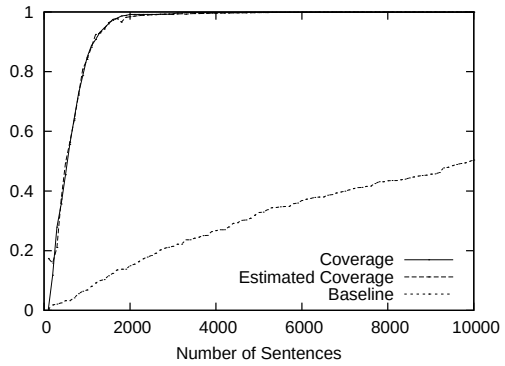

Figure 7: Annotation of RNA in the GENIA corpus.

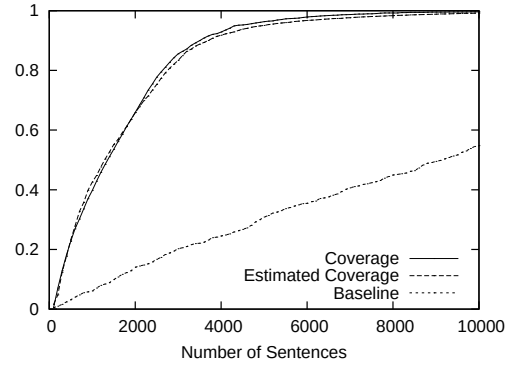

Figure 8: Annotation of cell_line in the GENIA corpus.

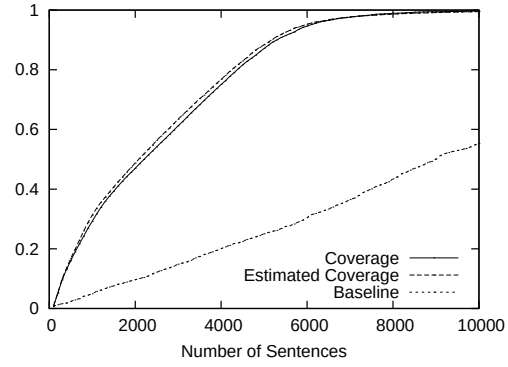

Figure 9: Annotation of cell_type in the GENIA corpus.

named entities of the category "MISC" are annotated.

We should also note that coverage estimation was surprisingly accurate. In all experiments, the difference between the estimated coverage and the real coverage was very small. This means that we can safely use the estimated coverage as the stopping condition for the annotation work.

Figures 6 to 9 show the experimental results on the GENIA data. The figures show the same characteristics observed in the CoNLL data. The acceleration by our framework was most evident for the "RNA" category.

Table 4 shows how much we can save the annotation cost if we stop the annotation process when the estimated coverage reaches 99%. The first column shows the coverage actually achieved and the second column shows the number and ratio of the sentences annotated in the corpus. This table shows that, on average, we can achieve a coverage of 99.0% by annotating 52.4% of the sentences in the corpus. In other words, we could roughly halve the annotation cost by accepting the missing rate of 1.0%.

As expected, the cost reduction was most drastic when "RNA", which is the most sparse named entity category (see Table 3), was targeted. The cost reduction was more than seven-fold. These experimental results confirm that our annotation framework is particularly useful when applied to sparse named entities.

Table 4 also shows the timing information on the experiments [5]. One of the potential problems with this kind of active learning-like framework is the computation time required to retrain the tagger at each iteration. Since the human annotator has to wait while the tagger is being retrained, the computation time required for retraining the tagger should not be very long. In our experiments, the worst case (i.e. DNA) required 443 seconds for retraining the tagger at the last iteration, but in most cases

[5] We used AMD Opteron 2.2GHz servers for the experiments and our CRF tagger is implemented in C++.

	Coverage	Sentences Annotated (%)	Cumulative Time (second)	Last Interval (second)
CoNLL: LOC	99.1%	7,600 (54.1%)	3,362	92
CoNLL: MISC	96.9%	5,400 (38.5%)	1,818	61
CoNLL: ORG	99.7%	8,900 (63.4%)	5,201	104
CoNLL: PER	98.0%	6,200 (44.2%)	2,300	75
GENIA: DNA	99.8%	11,900 (64.2%)	33,464	443
GENIA: RNA	99.2%	2,500 (13.5%)	822	56
GENIA: cell_line	99.6%	9,400 (50.7%)	15,870	284
GENIA: cell_type	99.3%	8,600 (46.4%)	13,487	295
Average	99.0%	- (52.4%)	-	-

Table 4: Coverage achieved when the estimated coverage reached 99%.

the training time for each iteration was kept under several minutes.

In this work, we used the BFGS algorithm for training the CRF model, but it is probably possible to further reduce the training time by using more recent parameter estimation algorithms such as exponentiated gradient algorithms (Globerson et al., 2007).

6 Discussion and Related Work

Our annotation framework is, by definition, not something that can ensure a coverage of 100%. The seriousness of a missing rate of, for example, 1% is not entirely clear—it depends on the application and the purpose of annotation. In general, however, it is hard to achieve a coverage of 100% in real annotation work even if the human annotator scans through all sentences, because there is often ambiguity in deciding whether a particular named entity should be annotated or not. Previous studies report that inter-annotator agreement rates with regards to gene/protein name annotation are f-scores around 90% (Morgan et al., 2004; Vlachos and Gasperin, 2006). We believe that the missing rate of 1% can be an acceptable level of sacrifice, given the cost reduction achieved and the unavoidable discrepancy made by the human annotator.

At the same time, we should also note that our framework could be used in conjunction with existing methods for semi-supervised learning to improve the performance of the CRF tagger, which in turn will improve the coverage. It is also possible to improve the performance of the tagger by using external dictionaries or using more sophisticated probabilistic models such as semi-Markov CRFs (Sarawagi and Cohen, 2004). These enhancements should further improve the coverage, keeping the same degree of cost reduction.

The idea of improving the efficiency of annotation work by using automatic taggers is certainly not new. Tanabe et al. (2005) applied a gene/protein name tagger to the target sentences and modified the results manually. Culotta and McCallum (2005) proposed to have the human annotator select the correct annotation from multiple choices produced by a CRF tagger for each sentence. Tomanek et al. (2007) discuss the reusability of named entity-annotated corpora created by an active learning approach and show that it is possible to build a corpus that is useful to different machine learning algorithms to a certain degree.

The limitation of our framework is that it is useful only when the target named entities are sparse because the upper bound of cost saving is limited by the proportion of the relevant sentences in the corpus. Our framework may therefore not be suitable for a situation where one wants to make annotations for named entities of many categories simultaneously (e.g. creating a corpus like GENIA from scratch). In contrast, our framework should be useful in a situation where one needs to modify or enrich named entity annotations in an existing corpus, because the target named entities are almost always sparse in such cases. We should also note that named entities in full papers, which recently started to attract much attention, tend to be more sparse than those in abstracts.

7 Conclusion

We have presented a simple but powerful framework for reducing the human effort for making name entity annotations in a corpus. The proposed framework allows us to annotate *almost* all named entities

of the target category in the given corpus without having to scan through all the sentences. The framework also allows us to know when to stop the annotation process by consulting the estimated coverage of annotation.

Experimental results demonstrated that the framework can reduce the number of sentences to be annotated almost by half, achieving a coverage of 99.0%. Our framework was particularly effective when the target named entities were very sparse.

Unlike active learning, this work enables us to create a named entity corpus that is free from the sampling bias introduced by the active learning strategy. This work will therefore be especially useful when one needs to enrich an existing linguistic corpus (e.g. WSJ, GENIA, or PennBioIE) with named entity annotations for a new semantic category.

Acknowledgment

This work is partially supported by BBSRC grant BB/E004431/1. The UK National Centre for Text Mining is sponsored by the JISC/BBSRC/EPSRC.

References

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Proceedings of AAAI-05*, pages 746–751.

Ido Dagan and Sean P. Engelson. 1995. Committee-based sampling for training probabilistic classifiers. In *Proceedings of ICML*, pages 150–157.

Sean Engelson and Ido Dagan. 1996. Minimizing manual annotation cost in supervised training from corpora. In *Proceedings of ACL*, pages 319–326.

A. Globerson, T. Koo, X. Carreras, and M. Collins. 2007. Exponentiated gradient algorithms for log-linear structured prediction. In *Proceedings of ICML*, pages 305–312.

J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. 2003. GENIA corpus–a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 (Suppl. 1):180–182.

J.-D. Kim, T. Ohta, Y. Tsuruoka, Y. Tateisi, and N. Collier. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA)*, pages 70–75.

Seth Kulick, Ann Bies, Mark Libeman, Mark Mandel, Ryan McDonald, Martha Palmer, Andrew Schein, and Lyle Ungar. 2004. Integrated annotation for biomedical information extraction. In *Proceedings of HLT-NAACL 2004 Workshop: Biolink 2004*, pages 61–68.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

Alexander A. Morgan, Lynette Hirschman, Marc Colosimo, Alexander S. Yeh, and Jeff B. Colombe. 2004. Gene name identification and normalization using a model organism database. *Journal of Biomedical Informatics*, 37:396–410.

Jorge Nocedal. 1980. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782.

Daisuke Okanohara, Yusuke Miyao, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2006. Improving the scalability of semi-markov conditional random fields for named entity recognition. In *Proceedings of COLING/ACL*, pages 465–472.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Proceedings of NIPS*.

Burr Settles. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In *COLING 2004 International Joint workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP) 2004*, pages 107–110.

Dan Shen, Jie Zhang, Jian Su, Guodong Zhou, and Chew-Lim Tan. 2004. Multi-criteria-based active learning for named entity recognition. In *Proceedings of ACL*, pages 589–596, Barcelona, Spain.

Lorraine Tanabe, Natalie Xie, Lynne H. Thom, Wayne Matten, and W. John Wilbur. 2005. GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6(Suppl 1):S3.

Cynthia A. Thompson, Mary Elaine Califf, and Raymond J. Mooney. 1999. Active learning for natural language parsing and information extraction. In *Proceedings of ICML*, pages 406–414.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147.

Katrin Tomanek, Joachim Wermter, and Udo Hahn. 2007. An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data. In *Proceedings of EMNLP-CoNLL*, pages 486–495.

Andreas Vlachos and Caroline Gasperin. 2006. Bootstrapping and evaluating named entity recognition in the biomedical domain. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 138–145.

The BioScope corpus: annotation for negation, uncertainty and their scope in biomedical texts

György Szarvas[1], Veronika Vincze[1], Richárd Farkas[2] and János Csirik[2]
[1]Department of Informatics
University of Szeged
H-6720, Szeged, Árpád tér 2.
[2]Research Group on Artificial Intelligence
Hungarian Academy of Science
H-6720, Szeged, Aradi vértanúk tere 1.
{szarvas, vinczev, rfarkas, csirik}@inf.u-szeged.hu

Abstract

This article reports on a corpus annotation project that has produced a freely available resource for research on handling negation and uncertainty in biomedical texts (we call this corpus the BioScope corpus). The corpus consists of three parts, namely medical free texts, biological full papers and biological scientific abstracts. The dataset contains annotations at the token level for negative and speculative keywords and at the sentence level for their linguistic scope. The annotation process was carried out by two independent linguist annotators and a chief annotator – also responsible for setting up the annotation guidelines – who resolved cases where the annotators disagreed. We will report our statistics on corpus size, ambiguity levels and the consistency of annotations.

1 Introduction

Detecting uncertain and negative assertions is essential in most Text Mining tasks where in general, the aim is to derive factual knowledge from textual data. This is especially so for many tasks in the biomedical (medical and biological) domain, where these language forms are used extensively in textual documents and are intended to express impressions, hypothesised explanations of experimental results or negative findings. Take, for example, the clinical coding of medical reports, where the coding of a negative or uncertain disease diagnosis may result in an over-coding financial penalty. Another example from the biological domain is interaction extraction, where the aim is to mine text evidence for biological entities with certain relations between them. Here, while an uncertain relation or the non-existence of a relation might be of some interest for an end-user as well, such information must not be confused with real textual evidence (reliable information). A general conclusion is that for text mining, extracted information that is within the scope of some negative / speculative (hedge or soft negation) keyword should either be discarded or presented separately from factual information.

Even though many successful text processing systems (Friedman et al., 1994, Chapman et al. 2001, Elkin et al. 2005) handle the above-mentioned phenomena, most of them exploit hand-crafted rule-based negation/uncertainty detection modules. To the best of our knowledge, there are no publicly available standard corpora of reasonable size that are usable for evaluating the automatic detection and scope resolution of these language phenomena. The availability of such a resource would undoubtedly facilitate the development of corpus-based statistical systems for negation/hedge detection and resolution.

Our study seeks to fill this gap by presenting the BioScope corpus, which consists of medical and biological texts annotated for negation, speculation and their linguistic scope. This was done to permit a comparison between and to facilitate the development of systems for negation/hedge detection and scope resolution. The corpus described in this paper has been made publicly available for research purposes and it is freely downloadable[1].

[1] www.inf.u-szeged.hu/rgai/bioscope

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 38–45,
Columbus, Ohio, USA, June 2008.

1.1 Related work

Chapman et al. (2001) created a simple regular expression algorithm called NegEx that can detect phrases indicating negation and identify medical terms falling within the negative scope. With this process, a large part of negatives can be identified in discharge summaries.

Mutalik et al. (2001) earlier developed Negfinder in order to recognise negated patterns in medical texts. Their lexer uses regular expressions to identify words indicating negation and then it passes them as special tokens to the parser, which makes use of the single-token look-ahead strategy. Thus, without appealing to the syntactic structure of the sentence, Negfinder can reliably identify negated concepts in medical narrative when they are located near the negation markers.

Huang and Lowe (2007) implemented a hybrid approach to automated negation detection. They combined regular expression matching with grammatical parsing: negations are classified on the basis of syntactic categories and they are located in parse trees. Their hybrid approach is able to identify negated concepts in radiology reports even when they are located at some distance from the negative term.

The Medical Language Extraction and Encoding (MedLEE) system was developed as a general natural language processor in order to encode clinical documents in a structured form (Friedman et al., 1994). Negated concepts and certainty modifiers are also encoded within the system, thus it enables them to make a distinction between negated/uncertain concepts and factual information which is crucial in information retrieval.

Elkin et al. (2005) use a list of negation words and a list of negation scope-ending words in order to identify negated statements and their scope.

Although a fair amount of literature on uncertainty (or hedging) in scientific texts has been produced since the 1990s (e.g. Hyland, 1994), speculative language from a Natural Language Processing perspective has only been studied in the past few years. Previous studies (Light et al., 2004) showed that the detection of hedging can be solved effectively by looking for specific keywords which imply speculative content.

Another possibility is to treat the problem as a classification task and train a statistical model to discriminate speculative and non-speculative assertions. This approach requires the availability of labeled instances to train the models on. Medlock and Briscoe (2007) proposed a weakly supervised setting for hedge classification in scientific texts where the aim is to minimise human supervision needed to obtain an adequate amount of training data. Their system focuses on locating hedge cues in text and thus they do not determine the scopes (in other words in a text they define the scope to be a whole sentence).

1.2 Related resources

Even though the problems of negation (mainly in the medical domain) and hedging (mainly in the scientific domain) have received much interest in the past few years, open access annotated resources for training, testing and comparison are rare and relatively small in size. Our corpus is the first one with an annotation of negative/speculative keywords and their scope. The authors are only aware of the following related corpora:

- The ***Hedge classification corpus*** (Medlock and Briscoe, 2007), which has been annotated for hedge cues (at the sentence level) and consists of five full biological research papers (1537 sentences). No scope annotation is given in the original corpus. We included this publicly available corpus in ours, enriching the data with annotation for negation cues and linguistic scope for both hedging and negation.
- The ***Genia Event corpus*** (Kim et al., 2008), which annotates biological events with negation and three levels of uncertainty (1000 abstracts).
- The ***BioInfer corpus*** (Pyysalo et al., 2007), where biological relations are annotated for negation (1100 sentences in size).

In the two latter corpora biological terms (relations and events) have been annotated for both negation and hedging, but linguistic cues (i.e. which keyword modifies the semantics of the statement) have not been annotated. We annotated keywords and their linguistic scope, which is very useful for machine learning or rule-based negation and hedge detection systems.

2 Annotation guidelines

This section describes the basic principles on the annotation of speculative and negative scopes in biomedical texts. Some basic definitions and technical details are given in Section 2.1, then the general guidelines are discussed in Section 2.2 and the most typical keywords and their scopes are illustrated with examples in Section 2.3. Some special cases and exceptions are listed in Section 2.4, then the annotation process of the corpus is described and discussed in Section 2.5. The complete annotation guidelines document is available from the corpus homepage.

2.1 Basic issues

In a text, just sentences with some instance of speculative or negative language are considered for annotation. The annotation is based on linguistic principles, i.e. parts of sentences which do not contain any biomedical term are also annotated if they assert the non-existence/uncertainty of something.

As for speculative annotation, if a sentence is a statement, that is, it does not include any speculative element that suggests uncertainty, it is disregarded. Questions inherently suggest uncertainty – which is why they are asked –, but they will be neglected and not annotated unless they contain speculative language.

Sentences containing any kind of negation are examined for negative annotation. Negation is understood as the implication of the non-existence of something. However, the presence of a word with negative content does not imply that the sentence should be annotated as negative, since there are sentences that include grammatically negative words but have a speculative meaning or are actually regular assertions (see the examples below).

In the corpus, instances of speculative and negative language – that is, keywords and their scope – are annotated. Speculative elements are marked by angled brackets: <*or*>, <*suggests*> etc., while negative keywords are marked by square brackets: [*no*], [*without*] etc. The scope of both negative and speculative keywords is denoted by parentheses. Also, the speculative or negative cue is always included within its scope:

This result (<suggests> that the valency of Bi in the material is smaller than + 3).

Stable appearance the right kidney ([without] hydronephrosis).

In the following, the general guidelines for speculative and negative annotation are presented.

2.2 General guidelines

During the annotation process, we followed a *min-max* strategy for the marking of keywords and their scope. When marking the keywords, a minimalist strategy was followed: the minimal unit that expressed hedging or negation was marked as a keyword. However, there are some cases when hedge or negation can be expressed via a phrase rather than a single word. Complex keywords are phrases that express uncertainty or negation together, but they cannot do this on their own (the meaning or the semantics of its subcomponents are significantly different from the semantics of the whole phrase). An instance of a complex keyword can be seen in the following sentence:

Mild bladder wall thickening (<raises the question of> cystitis).

On the other hand, a sequence of words cannot be marked as a complex keyword if it is only one of those words that express speculative or negative content (even without the other word). Thus prepositions, determiners, adverbs and so on are not annotated as parts of the complex keyword if the keyword can have a speculative or negative content on its own:

The picture most (<likely> reflects airways disease).

Complex keywords are not to be confused with the sequence of two or more keywords because they can express hedge or negation on their own, that is, without the other keyword as well. In this case, each keyword is annotated separately, as is shown in the following example:

Slightly increased perihilar lung markings (<may> (<indicate> early reactive airways disease)).

2.3 Scope marking

When marking the scopes of negative and speculative keywords, we extended the scope to the biggest syntactic unit possible (in contrast to other corpora like the one described in (Mutalik et al., 2001)). Thus, annotated scopes always have the

maximal length – as opposed to the strategy for annotating keywords, where we marked the minimal unit possible. Our decision was supported by two facts. First, since scopes must contain their keywords, it seemed better to include every element in between the keyword and the target word in order to avoid "empty" scopes, that is, scopes without a keyword. In the next example, *however* is not affected by the hedge cue but it should be included within the scope, otherwise the keyword and its target phrase would be separated:

(Atelectasis in the right mid zone is, however, <possible>).

Second, the status of modifiers is occasionally vague: it is sometimes not clear whether the modifier of the target word belongs to its scope as well. The following sentence can describe two different situations:

There is [no] primary impairment of glucocorticoid metabolism in the asthmatics.

First, the glucocorticoid metabolism is impaired in the asthmatics but not primarily, that is, the scope of *no* extends to *primary*. Second, the scope of *no* extends to *impairment* (and its modifiers and complements as well), thus there is no impairment of the glucocorticoid metabolism at all. Another example is shown here:

Mild viral <or> reactive airways disease is detected.

The syntactic structure of the above sentence is ambiguous. First, the airways disease is surely mild, but it is not known whether it is viral or reactive; or second, the airways disease is either mild and viral or reactive and not mild. Most of the sentences with similar problems cannot be disambiguated on the basis of contextual information, hence the proper treatment of such sentences remains problematic. However, we chose to mark the widest scope available: in other words, we preferred to include every possible element within the scope rather than exclude elements that should probably be included.

The scope of a keyword can be determined on the basis of syntax. The scope of verbs, auxiliaries, adjectives and adverbs usually extends to the right of the keyword. In the case of verbal elements, i.e. verbs and auxiliaries, it ends at the end of the clause (if the verbal element is within a relative clause or a coordinated clause) or the sentence, hence all complements and adjuncts are included, in accordance with the principle of maximal scope size. Take the following examples:

The presence of urothelial thickening and mild dilatation of the left ureter (<suggest> that the patient may have continued vesicoureteral reflux).

These findings that (<may> be from an acute pneumonia) include minimal bronchiectasis as well.

These findings (<might> be chronic) and (<may> represent reactive airways disease).

The scope of attributive adjectives generally extends to the following noun phrase, whereas the scope of predicative adjectives includes the whole sentence. For example, in the following two statements:

This is a 3 month old patient who had (<possible> pyelonephritis) with elevated fever.

(The demonstration of hormone receptor proteins in cells from malignant effusions is <possible>).

Sentential adverbs have a scope over the entire sentence, while the scope of other adverbs usually ends at the end of the clause or sentence. For instance,

(The chimaeric oncoprotein <probably> affects cell survival rather than cell growth).

Right upper lobe volume loss and (<probably> pneumonia).

The scope of conjunctions extends to all members of the coordination. That is, it usually extends to the both left and right:

Symptoms may include (fever, cough <or> itches).

Complex keywords such as *either … or* have one scope:

Mild perihilar bronchial wall thickening may represent (<either> viral infection <or> reactive airways disease).

Prepositions have a scope over the following (noun) phrase:

Mildly hyperinflated lungs ([without] focal opacity).

When the subject of the sentence contains the negative determiners *no* or *neither*, its scope extends to the entire sentence:

Surprisingly, however, ([neither] of these proteins bound in vitro to EBS1 or EBS2).

The main exception that changes the original scope of the keyword is the passive voice. The subject of the passive sentence was originally the object of the verb, that is, it should be within its scope. This is why the subject must also be marked within the scope of the verb or auxiliary. For instance,

(A small amount of adenopathy <cannot be> completely <excluded>).

Another example of scope change is the case of raising verbs (*seem, appear, be expected, be likely* etc.). These can have two different syntactic patterns, as the following examples suggest:

It seems that the treatment is successful.

The treatment seems to be successful.

In the first case, the scope of *seems* starts right with the verb. If this was the case in the second pattern, *the treatment* would not be included in the scope, but it should be like that shown in the first pattern. Hence in the second sentence, the scope must be extended to the subject as well:

It (<seems> that the treatment is successful).

(The treatment <seems> to be successful).

Sometimes a negative keyword is present in the text apparently without a scope: *negative* obviously expresses negation, but the negated fact – what medical problem the radiograph is negative for – is not part of the sentence. In such cases, the keyword is marked and the scope contains just the keyword:

([Negative]) chest radiograph.

In the case of elliptic sentences, the same strategy is followed: the keyword is marked and its scope includes only the keyword since the verbal phrase, that is, the scope of *not*, is not repeated in the sentence.

This decrease was seen in patients who responded to the therapy as well as in those who did ([not]).

Generally, punctuation marks or conjunctions function as scope boundary markers in the corpus, in contrast to the corpus described in (Mutalik et al., 2001) where certain lexical items are treated as negation-termination tokens. Since in our corpus the scope of negation or speculation is mostly extended to the entire clause in the case of verbal elements, it is clear that markers of a sentence or clause boundary determine the end of their scope.

2.4 Special cases

It seems unequivocal that whenever there is a speculative or negative cue in the sentence, the sentence expresses hedge or negation. However, we have come across several cases where the presence of a speculative/negative keyword does not imply a hedge/negation. That is, some of the cues do not denote speculation or negation in all their occurrences, in other words, they are ambiguous.

For instance, the following sentence is a statement and it is the degree of probability that is precisely determined, but it is not an instance of hedging although it contains the cue *probable*:

The planar amide groups in which is still digging nylon splay around 30 less probable event.

As for negative cues, sentences including a negative keyword are not necessarily to be annotated for negation. They can, however, have a speculative content as well. The following sentence contains *cannot*, which is a negative keyword on its own, but not in this case:

(A small amount of adenopathy <cannot be> completely <excluded>).

Some other sentences containing a negative keyword are not to be annotated either for speculation or for negation. In the following example, the negative keyword is accompanied by an adverb and their meaning is neither speculative nor negative. The sequence of the negative keyword and the adverb can be easily substituted by another adverb or adjective having the same (or a similar) meaning, which is by no means negative – as shown in the example. In this way, the sentence below can be viewed as a positive assertion (not a statement of the non-existence of something).

Thus, signaling in NK3.3 cells is not always (=sometimes) identical with that in primary NK cells.

As can be seen from the above examples, hedging or negation is determined not just by the presence

of an apparent cue: it is rather an issue of the keyword, the context and the syntactic structure of the sentence taken together.

2.5 Annotation process

Our BioScope corpus was annotated by two independent linguists following the guidelines written by our linguist expert before the annotation of the corpus was initiated. These guidelines were developed throughout the annotation process as annotators were often confronted with problematic issues. The annotators were not allowed to communicate with each other as far as the annotation process was concerned, but they could turn to the expert when needed and regular meetings were also held between the annotators and the linguist expert in order to discuss recurring and/or frequent problematic issues. When the two annotations for one subcorpus were finalised, differences between the two were resolved by the linguist expert, yielding the gold standard labeling of the subcorpus.

3 Corpus details

In this section we will discuss in detail the overall characteristics of the corpus we developed, including a brief description of the texts that constitute the BioScope corpus and some general statistics concerning the size of each part, distribution of negation/hedge cues, ambiguity levels and finally we will present statistics on the final results of the annotation work.

3.1 Corpus texts

The corpus consists of texts taken from 4 different sources and 3 different types in order to ensure that it captures the heterogenity of language use in the biomedical domain. We decided to add clinical free-texts (radiology reports), biological full papers and biological paper abstracts (texts from Genia).

Table 1 summarises the chief characteristics of the three subcorpora. The 3rd and 5th rows of the table show the ratio of sentences which contain negated or uncertain statements. The 4rd and 6th rows show the number of negation and hedge cue occurrences in the given corpus.

A major part of the corpus consists of clinical free-texts. We chose to add medical texts to the corpus in order to facilitate research on negation/hedge detection in the clinical domain. The radiology report corpus that was used for the clinical coding challenge (Pestian et al., 2007) organised by the Computational Medicine Center in Cincinatti, Ohio in 2007 was annotated for negations and uncertainty along with the scopes of each phenomenon. This part contains 1954 documents, each having a *clinical history* and an *impression* part, the latter being denser in negated and speculative parts.

Another part of the corpus consists of full scientific articles. 5 articles from FlyBase (the same data were used by Medlock and Briscoe (2007) for evaluating sentence-level hedge classifiers) and 4 articles from the open access BMC Bioinformatics website were downloaded and annotated for negations, uncertainty and their scopes. Full papers are particularly useful for evaluating negation/hedge classifiers as different parts of an article display different properties in the use of speculative or negated phrases. Take, for instance, the *Conclusions* section of scientific papers that tends to contain significantly more uncertain or negative findings than the description of *Experimental settings and methods*.

Scientific abstracts are the main targets for various Text Mining applications like protein-protein interaction mining due to their public accessibility (e.g. through PubMed). We therefore decided to include quite a lot of texts from the abstracts of scientific papers. This is why we included the abstracts of the Genia corpus (Collier et al., 1999). This decision was straightforward for two reasons. First, the Genia corpus contains syntax tree annotation, which allows a comparison between scope annotation and syntactic structure. Being syntactic in nature, scopes should align with the bracket structure of syntax trees, while scope resolution algorithms that exploit treebank data can be used as a theoretical upper bound for the evaluation of parsers for resolving negative/hedge scopes. The other reason was that scope annotation can mutually benefit from the rich annotations of the Genia corpus, such as term annotation (evaluation) and event annotation (comparison with the biologist uncertainty labeling of events).

The corpus consists of more than 20.000 annotated sentences altogether. We consider this size to be sufficiently large to serve as a standard evaluation corpus for negation/hedge detection in the biomedical domain.

	Clinical	Full Paper	Abstract
#Documents	1954	9	1273
#Sentences	6383	2624	11872
Negation sentences	6.6%	13.76%	13.45%
#Negation cues	871	404	1757
Hedge sentences	13.4%	22.29%	17.69%
#Hedge cues	1137	783	2691

Table 1: Statistics of the three subcorpora.

3.2 Agreement analysis

We measured the consistency level of the annotation using inter-annotator agreement analysis. The inter-annotator agreement rate is defined as the $F_{ß=1}$ measure of one annotation, treating the second one as the gold standard. We calculated agreement rates for all three subcorpora between the two independent annotators and between the two annotators and the gold standard labeling. The gold standard labeling was prepared by the creator of the annotation guide, who resolved all cases where the two annotators disagreed on a keyword or its scope annotation.

We measured the agreement rate of annotating negative and hedge keywords, and the agreement rate of annotating the linguistic scope for each phenomenon. We distinguished left-scope, right-scope and full scope agreement that required both left and right scope boundaries to match exactly to be considered as coinciding annotations. A detailed analysis of the consistency levels for the three subcorpora and the ambiguity levels for each negative and hedge keyword (that is, the ratio of a keyword being annotated as a negative/speculative cue and the number of all the occurrences of the same keyword in the corpus) can be found at the corpus homepage.

3.3 BioScope corpus availability

The corpus is available free of charge for research purposes and can be obtained for a modest price for business use. For more details, see the BioScope homepage:
www.inf.u-szeged.hu/rgai/bioscope.

4 Conclusions

In this paper we reported on the construction of a corpus annotated for negations, speculations and their linguistic scopes. The corpus is accessible for academic purposes and is free of charge. Apart from the intended goal of serving as a common resource for the training, testing and comparison of biomedical Natural Language Processing systems, the corpus is also a good resource for the linguistic analysis of scientific and clinical texts.

The most obvious conclusions here are that the usual language of clinical documents makes it much easier to detect negation and uncertainty cues than in scientific texts because of the very high ratio of the actual cue words (i.e. low ambiguity level), which explains the high accuracy scores reported in the literature. In scientific texts – which are nowadays becoming a popular target for Text Mining (for literature-based knowledge discovery) – the detection and scope resolution of negation and uncertainty is, on the other hand, a problem of great complexity, with the percentage of non-hedge occurrences being as high as 90% for some hedge cue candidates in biological paper abstracts. Take for example the keyword *or* which is labeled as a speculative keyword in only 11.32% of the cases in scientific abstracts, while it was labeled as speculative in 97.86% of the cases in clinical texts. Identifying the scope is also more difficult in scientific texts where the average sentence length is much longer than in clinical data, and the style of the texts is also more literary in the former case.

In our study we found that hedge detection is a more difficult problem than identifying negations because the number of possible cue words is higher and the ratio of real cues is significantly lower in the case of speculation (higher keyword/non-keyword ambiguity). The annotator-agreement table also confirms this opinion: the detection of hedging is more complicated than negation even for humans.

Our corpus statistics also prove the importance of negation and hedge detection. The ratio of negated and hedge sentences in the corpus varies in the subcorpora, but we can say that over 20% of the sentences contains a modifier that radically influences the semantic content of the sentence.

One of the chief construction principles of the BioScope corpus was to facilitate the training/development of automatic negation and hedge detection systems. Such systems have to solve two sub-problems: they have to identify real cue words (note that the probability of any word being a keyword can be different for various domains) and

then they have to determine the linguistic scope of actual keywords.

These automatic hedge and negation detection methods can be utilised in a variety of ways in a (biomedical) Text Mining system. They can be used as a preprocessing tool, i.e. each word in a detected scope can be removed from the documents if we seek to extract true assertions. This can significantly reduce the level of noise for processing in such cases where only a document-level labeling is provided (like that for the ICD-9 coding dataset) and just clear textual evidence for certain things should be extracted. On the other hand, similar systems can classify previously extracted statements according to their certainty or uncertainty, which is generally an important issue in the automatic processing of scientific texts.

Acknowledgments

This work was supported in part by the NKTH grant of the Jedlik Ányos R&D Programme 2007 (project codename TUDORKA7) of the Hungarian government. The authors wish to thank the anonymous reviewers for their useful suggestions and comments. The authors also wish to thank the creators of the ICD-9 coding dataset and the Genia corpus for making the texts that were used here publicly available. The authors thank Jin-Dong Kim as well for the useful comments and suggestions on the annotation guide and Orsolya Vincze and Mihály Minkó (the two annotators) for their work.

References

Wendy W. Chapman, Will Bridewell, Paul Hanbury, Gregory F. Cooper and Bruce G. Buchanan. 2001. A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 34(5):301–310.

N. Collier, H. S. Park, N. Ogata, Y. Tateishi, C. Nobata, T. Ohta, T. Sekimizu, H. Imai, K. Ibushi, and J. Tsujii. 1999. The GENIA project: corpus-based knowledge acquisition and information extraction from genome research papers. *Proceedings of EACL-99*.

Peter L. Elkin, Steven H. Brown, Brent A. Bauer, Casey S. Husser, William Carruth, Larry R. Bergstrom and Dietlind L. Wahner-Roedler. 2005. A controlled trial of automated classification of negation from clinical notes. *BMC Medical Informatics and Decision Making* 5:13 doi:10.1186/1472-6947-5-13.

C. Friedman, P.O. Alderson, J.H. Austin, J.J. Cimino, and S.B. Johnson. 1994. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174.

Yang Huang and Henry J. Lowe. 2007. A Novel Hybrid Approach to Automated Negation Detection in Clinical Radiology Reports. *Journal of the American Medical Informatics Association*, 14(3):304–311.

Ken Hyland. 1994. *Hedging in academic writing and EAP textbooks. English for Specific Purposes*, 13(3):239–256.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics* 2008, 9:10.

Marc Light, Xin Ting Qui, and Padmini Srinivasan. 2004. The language of bioscience: Facts, speculations, and statements in between. In *Proceedings of BioLink 2004 Workshop on Linking Biological Literature, Ontologies and Databases: Tools for Users*. Boston, Massachusetts, Association for Computational Linguistics, 17–24.

Ben Medlock and Ted Briscoe. 2007. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Association for Computational Linguistics, 992–999.

Pradeep G. Mutalik, Aniruddha Deshpande, and Prakash M. Nadkarni. 2001. Use of General-purpose Negation Detection to Augment Concept Indexing of Medical Documents: A Quantitative Study Using the UMLS. *Journal of the American Medical Informatics Association*, 8(6):598–609.

John P. Pestian, Chris Brew, Pawel Matykiewicz, DJ Hovermale, Neil Johnson, K. Bretonnel Cohen, and Wlodzislaw Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Biological, translational, and clinical language processing*. Prague, Association for Computational Linguistics, 97–104.

Sampo Pyysalo, Filip Ginter, Juho Heimonen, Jari Björne, Jorma Boberg, Jouni Järvinen, and Tapio Salakoski. 2007. BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics* 2007, 8:50 doi:10.1186/1471-2105-8-50.

Recognizing Speculative Language in Biomedical Research Articles: A Linguistically Motivated Perspective

Halil Kilicoglu and Sabine Bergler
Department of Computer Science and Software Engineering
Concordia University
Montreal, Quebec, Canada
{h_kilico, bergler}@cse.concordia.ca

Abstract

We explore a linguistically motivated approach to the problem of recognizing speculative language ("hedging") in biomedical research articles. We describe a method, which draws on prior linguistic work as well as existing lexical resources and extends them by introducing syntactic patterns and a simple weighting scheme to estimate the speculation level of the sentences. We show that speculative language can be recognized successfully with such an approach, discuss some shortcomings of the method and point out future research possibilities.

1 Introduction

Science involves making hypotheses, experimenting, and reasoning to reach conclusions, which are often tentative and provisional. Scientific writing, particularly in biomedical research articles, reflects this, as it is rich in speculative statements, also known as *hedges*. Most text processing systems ignore hedging and focus on factual language (assertions). Although assertions, sometimes mere co-occurrence of terms, are the focus of most information extraction and text mining applications, identifying hedged text is crucial, because hedging alters, in some cases even reverses, factual statements. For instance, the italicized fragment in example (1) below implies a factual statement while example (2) contains two hedging cues (*indicate* and *might*), which render the factual proposition speculative:

(1) Each empty cell *indicates that* the corresponding TPase query was not used at the particular stage of PSI-BLAST analysis.

(2) These experiments indicated that the roX genes might function as nuclear entry sites for the assembly of the MSL proteins on the X chromosome.

These examples not only illustrate the phenomenon of hedging in the biomedical literature, they also highlight some of the difficulties in recognizing hedges. The word *indicate* plays a different role in each example, acting as a hedging cue only in the second.

In recent years, there has been increasing interest in the speculative aspect of biomedical language (Light et al., 2004, Wilbur et al., 2006, Medlock and Briscoe, 2007). In general, these studies focus on issues regarding annotating speculation and approach the problem of recognizing speculation as a text classification problem, using the well-known "bag of words" method (Light et al, 2004, Medlock and Briscoe, 2007) or simple substring matching (Light et al., 2004). While both approaches perform reasonably well, they do not take into account the more complex and strategic ways hedging can occur in biomedical research articles. In example (3), hedging is achieved with a combination of referring to experimental results (*We ... show that ... indicating)* and the prepositional phrase *to our knowledge*:

(3) We further show that D-mib is specifically required for Ser endocytosis and signaling during wing development indicating for the first time to our knowledge that endocytosis regulates Ser signaling.

In this paper, we extend previous work through linguistically motivated techniques. In particular, we pay special attention to syntactic structures. We

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 46–53,
Columbus, Ohio, USA, June 2008.

address lexical hedges by drawing on a set of lexical hedging cues and expanding and refining it in a semi-automatic manner to acquire a hedging dictionary. To capture more complex strategic hedges, we determine syntactic patterns that commonly act as hedging indicators by analyzing a publicly available hedge classification dataset. Furthermore, recognizing that "not all hedges are created equal", we use a weighting scheme, which also takes into consideration the strengthening or weakening effect of certain syntactic structures on lexical hedging cues. Our results demonstrate that linguistic knowledge can be used effectively to enhance the understanding of speculative language.

2 Related Work

The term *hedging* was first used in linguistic context by Lakoff (1972). He proposed that natural language sentences can be true or false to some extent, contrary to the dominant truth-conditional semantics paradigm of the era. He was mainly concerned with how words and phrases, such as *mainly* and *rather*, make sentences fuzzier or less fuzzy.

Hyland (1998) provides one of the most comprehensive accounts of hedging in scientific articles in the linguistics literature. He views hedges as polypragmatic devices with an array of purposes such as weakening the force of statement, expressing deference to the reader and signaling uncertainty. He proposes a fuzzy model, in which he categorizes scientific hedges by their pragmatic purpose, such as reliability hedges and reader-oriented hedges. He also identifies the principal syntactic realization devices for different types of hedges, including epistemic verbs (verbs indicating the speaker's mode of knowing), adverbs and modal auxiliaries and presents the most frequently used members of these types based on analysis of a molecular biology article corpus.

Palmer (1986) identifies epistemic modality, which expresses the speaker's degree of commitment to the truth of proposition and is closely linked to hedging. He identifies three types of epistemic modality: "speculatives" express uncertainty, "deductives" indicate an inference from observable evidence, and "assumptives" indicate inference from what is generally known. He focuses mainly on the use of modal verbs in expressing various types of epistemic modality.

In their investigation of event recognition in news text, Saurí et al. (2006) address event modality at the lexical and syntactic level by means of SLINKs (subordination links), some of which ("modal", "evidential") indicate hedges. They use corpus-induced lexical knowledge from TimeBank (Pustejovsky et al. (2003)), standard linguistic predicate classifications, and rely on a finite-state syntactic module to identify subordinated events based on the subcategorization properties of the subordinating event.

DiMarco and Mercer (2004) study the intended communicative purpose (dispute, confirmation, use of materials, tools, etc.) of citations in scientific text and show that hedging is used more frequently in citation contexts.

In the medical field, Friedman et al. (1994) discuss uncertainty in radiology reports and their natural language processing system assigns one of five levels of certainty to extracted findings.

Light et al. (2004) explore issues with annotating speculative language in biomedicine and outline potential applications. They manually annotate a corpus of approximately 2,000 sentences from MEDLINE abstracts. Each sentence is annotated as being definite, low speculative and highly speculative. They experiment with simple substring matching and a SVM classifier, which uses single words as features. They obtain slightly better accuracy with simple substring matching suggesting that more sophisticated linguistic knowledge may play a significant role in identification of speculative language. It is also worth noting that both techniques yield better accuracy over full abstracts than on the last two sentences of abstracts, in which speculative language is found to be more prevalent.

Medlock and Briscoe (2007) extend Light et al.'s (2004) work, taking full-text articles into consideration and applying a weakly supervised learning model, which also uses single words as features, to classify sentences as simply speculative or non-speculative. They manually annotate a test set and employ a probabilistic model for training set acquisition using *suggest* and *likely* as seed words. They use Light et al.'s substring matching as the baseline and improve to a recall/precision break-even point (BEP) of 0.76, using a SVM committee-based model from 0.60 recall/precision BEP of the baseline. They note that their learning models are unsuccessful in identify-

ing assertive statements of knowledge paucity, generally marked syntactically rather than lexically.

Wilbur et al. (2006) suggest that factual information mining is not sufficient and present an annotation scheme, in which they identify five qualitative dimensions that characterize scientific sentences: focus (generic, scientific, methodology), evidence (E0-E3), certainty (0-3), polarity (positive, negative) and trend (+,-). Certainty and evidence dimensions, in particular, are interesting in terms of hedging. They present this annotation scheme as the basis for a corpus that will be used to automatically classify biomedical text.

Discussion of hedging in Hyland (1998) provides the basic linguistic underpinnings of the study presented here. Our goals are similar to those outlined in the work of Light et al. (2004) and Medlock and Briscoe (2007); however, we propose that a more linguistically oriented approach not only could enhance recognizing speculation, but would also bring us closer to characterizing the semantics of speculative language. Some of the work discussed above (in particular, Saurí et al. (2006) and Wilbur et al. (2006)) will be relevant in that regard.

3 Methods

To develop an automatic method to identify speculative sentences, we first compiled a set of core lexical surface realizations of hedging drawn from Hyland (1998). Next, we augmented this set by analyzing a corpus of 521 sentences, 213 of which are speculative, and also noted certain syntactic structures used for hedging. Furthermore, we identified lexical cues and syntactic patterns that strongly suggest non-speculative contexts ("unhedgers"). We then expanded and manually refined the set of lexical hedging and "unhedging" cues using WordNet (Fellbaum, 1998) and the UMLS SPECIALIST Lexicon (McCray et al., 1994). Next, we quantified the strength of the hedging cues and patterns through corpus analysis. Finally, to recognize the syntactic patterns, we used the Stanford Lexicalized Parser (Klein and Manning, 2003) and its dependency parse representation (deMarneffe et al., 2006). We use weights assigned to hedging cues to compute an overall hedging score for each sentence.

To evaluate the effectiveness of our method, we used basic information retrieval evaluation metrics: precision, recall, accuracy and F_1 score. In addition, we measure the recall/precision break-even point (BEP), which indicates the point at which precision and recall are equal, to provide a comparison to results previously reported. As baseline, we use the substring matching method, described in Light et al. (2004) in addition to another substring matching method, which uses terms ranked in top 15 in Medlock and Briscoe (2007). To measure the statistical significance of differences between the performances of baseline and our system, we used the binomial sign test.

4 Data Set

In our experiments, we use the publicly available hedge classification dataset[1], reported in Medlock and Briscoe (2007). This dataset consists of a manually annotated test set of 1537 sentences (380 speculative) extracted from six full-text articles on *Drosophila melanogaster* (fruit-fly) and a training set of 13,964 sentences (6423 speculative) automatically induced using a probabilistic acquisition model. A pool of 300,000 sentences randomly selected from an archive of 5579 full-text articles forms the basis for training data acquisition and drives their weakly supervised hedge classification approach.

While this probabilistic model for training data acquisition is suitable for the type of weakly supervised learning approach they describe, we find that it may not be suitable as a fair data sample, since the speculative instances overemphasize certain hedging cues used as seed terms (*suggest, likely*). On the other hand, the manually annotated test set is valuable for our purposes. To train our system, we (the first author) manually annotated a separate training set of 521 sentences (213 speculative) from the pool, using the annotation guidelines provided. Despite being admittedly small, the training set seems to provide a good sample, as the distribution of surface realization features (epistemic verbs (32%), adverbs (26%), adjectives (19%), modal verbs (%21)) correspond roughly to that presented in Hyland (1998).

5 Core Surface Realizations of Hedging

[1] http://www.benmedlock.co.uk/hedgeclassif.html

Hyland (1998) provides the most comprehensive account of surface realizations of hedging in scientific articles, categorizing them into two classes: lexical and non-lexical features. Lexical features include modal auxiliaries (*may* and *might* being the strongest indicators), epistemic verbs, adjectives, adverbs and nouns. Some common examples of these feature types are given in Table 1.

Feature Type	Examples
Modal auxiliaries	*may, might, could, would, should*
Epistemic judgment verbs	*suggest, indicate, speculate, believe, assume*
Epistemic evidential verbs	*appear, seem*
Epistemic deductive verbs	*conclude, infer, deduce*
Epistemic adjectives	*likely, probable, possible*
Epistemic adverbs	*probably, possibly, perhaps, generally*
Epistemic nouns	*possibility, suggestion*

Table 1. Lexical surface features of hedging

Non-lexical hedges usually include reference to limiting experimental conditions, reference to a model or theory or admission to a lack of knowledge. Their surface realizations typically go beyond words and even phrases. An example is given in sentence (4), with hedging cues italicized.

(4) Whereas much attention has focused on elucidating basic mechanisms governing axon development, *relatively little is known* about the genetic programs required for the establishment of dendrite arborization patterns that are hallmarks of distinct neuronal types.

While lexical features can arguably be exploited effectively by machine learning approaches, automatic identification of non-lexical hedges automatically seems to require syntactic and, in some cases, semantic analysis of the text.

Our first step was to expand on the core lexical surface realizations identified by Hyland (1998).

6 Expansion of Lexical Hedging Cues

Epistemic verbs, adjectives, adverbs and nouns provide the bulk of the hedging cues. Although epistemic features are commonly referred to and analyzed in the linguistics literature and various widely used lexicons exist that classify different part-of-speech (e.g., VerbNet (Kipper Schuler, 2005) for verb classes), we are unaware of any such comprehensive classification based on epistemological status of the words. We explore inducing such a lexicon from the core lexical examples identified in Hyland (1998) (a total of 63 hedging cues) and expanding it semi-automatically using two lexicons: WordNet (Fellbaum, 1998) and UMLS SPECIALIST Lexicon (McCray, 1994).

We first extracted synonyms for each epistemic term in our list using WordNet synsets. We then removed those synonyms that did not occur in our pool of sentences, since they are likely to be very uncommon words in scientific articles. Expanding epistemic verbs is somewhat more involved than expanding other epistemic types, as they tend to have more synsets, indicating a greater degree of word sense ambiguity (*assume* has 9 synsets). Based on the observation that an epistemic verb taking a clausal complement marked with *that* is a very strong indication of hedging, we only considered verb senses which subcategorize for a *that* complement. Expansion via WordNet resulted in 66 additional lexical features.

Next, we considered the case of nominalizations. Again, based on corpus analysis, we noted that nominalizations of epistemic verbs and adjectives are a common and effective means of hedging in molecular biology articles. The UMLS SPECIALIST Lexicon provides syntactic information, including nominalizations, for biomedical as well as general English terms. We extracted the nominalizations of words in our expanded dictionary of epistemic verbs and adjectives from UMLS SPECIALIST Lexicon and discarded those that do not occur in our pool of sentences, resulting in an additional 48 terms. Additional 5 lexical hedging cues (e.g., *tend, support*) were identified via manual corpus analysis and further expanded using the methodology described above.

An interesting class of cues are terms expressing strong certainty ("unhedgers"). Used within the scope of negation, these terms suggest hedging, while in the absence of negation they strongly suggest a non-speculative context. Examples of these include verbs indicating certainty, such as *know, demonstrate, prove and show*, and adjectives, such as *clear*. These features were also added to the dictionary and used together with other surface

cues to recognize speculative sentences. The hedging dictionary contains a total of 190 features.

7 Quantifying Hedging Strength

It is clear that not all hedging devices are equally strong and that the choice of hedging device affects the strength of the speculation. However, determining the strength of a hedging device is not trivial. The fuzzy pragmatic model proposed by Hyland (1998) employs general descriptive terms such as "strong" and "weak" when discussing particular cases of hedging and avoids the need for precise quantification. Light et al. (2004) report low inter-annotator agreement in distinguishing low speculative sentences from highly speculative ones. From a computational perspective, it would be useful to quantify hedging strength to determine the confidence of the author in his or her proposition.

As a first step in accommodating noticeable differences in strengths of hedging features, we assigned weights (1 to 5, 1 representing the lowest hedging strength and 5 the highest) to all hedging features in our dictionary. Core features were assigned weights based on the discussion in Hyland (1998). For instance, he identifies modal auxiliaries, *may* and *might*, as the prototypical hedging devices, and they were given weights of 5. On the other hand, modal auxiliaries commonly used in non-epistemic contexts (*would*, *could*) were assigned a lower weight of 3. Though not as strong as *may* and *might*, core epistemic verbs and adverbs are generally good hedging cues and therefore were assigned weights of 4. Core epistemic adjectives and nouns often co-occur with other syntactic features to act as strong hedging cues and were assigned weights of 3. Terms added to the dictionary via expansion were assigned a weight one less than their seed terms. For instance, the nominalization *supposition* has weight 2, since it is expanded from the verb suppose (weight 3), which is further expanded from its synonym speculate (weight 4), a core epistemic verb. The reduction in weights of certain hedging cues reflects their peripheral nature in hedging.

Hyland (1998) notes that writers tend to combine hedges ("harmonic combinations") and suggests the possibility of constructing scales of certainty and tentativeness from these combinations. In a similar vein, we accumulate the weights of the hedging features found in a sentence and assign an overall hedging score to each sentence.

8 The Role of Syntax

Corpus analysis shows that various syntactic devices play a prominent role in hedging, both as hedging cues and for strengthening or weakening effects. For instance, while some epistemic verbs do not act as hedging cues (or may be weak hedging cues) when used alone, together with a *that* complement or an infinitival clause, they are good indicators of hedging. A good example is *appear*, which often occurs in molecular biology articles with its "come into sight" meaning (5) and becomes a good hedging cue when it takes an infinitival complement (6):

(5) The linearity of the ommatidial arrangement was disrupted and numerous gaps appeared between ommatidia arrow.

(6) In these data a substantial fraction of both silent and replacement DNA mutations appear to affect fitness.

On the other hand, as discussed above, words expressing strong certainty ("unhedgers") are good indicators of hedging when negated, and strongly non-speculative otherwise.

We examined the training set and identified the most salient syntactic patterns that play a role in hedging. A syntactic pattern, or lack thereof, affects the overall score assigned to a hedging cue; a strengthening syntactic pattern will increase the overall score contributed by the cue, while a weakening pattern will decrease it. For instance, in sentence (5) above, the absence of the infinitival complement will reduce the score contribution of appear by 1, resulting in a score of 3 instead of 4. On the other hand, that appear takes an infinitival clause in example (6) will increase the score contribution of *appear* by 1. All score contributions of a sentence add up to its hedging score.

A purely syntactic case is that of *whether (if)*. Despite being a conjunction, it seems to act as a hedging cue when it introduces a clausal complement regardless of existence of any other hedging cue from the hedging dictionary. The basic syntactic patterns we identified and implemented and their effect on the overall hedging score are given in Table 2.

To obtain the syntactic structures of sentences, we used the statistical Stanford Lexicalized Parser (Klein and Manning, 2003), which provides a full parse tree, in addition to part-of-speech tagging based on the Penn Treebank tagset. A particularly useful feature of the Stanford Lexicalized Parser is typed dependency parses extracted from phrase structure parses (deMarneffe, et al. (2006)). We use these typed dependency parses to identify clausal complements, infinitival clauses and negation. For instance, the following two dependency relations indicate a clausal complement marked with *that* and identify the second syntactic pattern in Table 2.

ccomp(<EPISTEMIC VERB>,<VB>)
complm(<VB>,that)

In these relations, *ccomp* stands for clausal complement with internal subject and *complm* stands for complementizer. *VB* indicates any verb.

Syntactic Pattern	**Effect on Score**
<EPISTEMIC VERB> *to(inf)* VB	+1
<EPISTEMIC VERB> *that(comp)* VB	+2
Otherwise	-1
<EPISTEMIC NOUN> followed by *that(comp)*	+2
Otherwise	-1
not <UNHEDGING VERB>	+1
no| *not* <UNHEDGING NOUN>	+2
no| *not* immediately followed by <UNHEDGING ADVERB>	+1
no| *not* immediately followed by <UNHEDGING ADJECTIVE>	+1
whether| *if* in a clausal complement context	3

Table 2. Syntactic patterns and their effect on the overall hedging score.

9 Baseline

For our experiments, we used two baselines. First, we used the substring matching method reported in Light et al. (2004), which labels sentences containing one of more of the following as speculative: *suggest, potential, likely, may, at least, in part, possibl, further investigation, unlikely, putative, insights, point toward, promise* and *propose* (Baseline1). Secondly, we used the top 15 ranked term features determined using $P(spec|x_j)$ in training and classification models (at smoothing parameter α=5) reported in Medlock and Briscoe (2007): *suggest, likely, may, might, seems, Taken, suggests, probably, Together, suggesting, possibly, suggested, findings, observations, Given*. Our second baseline uses the substring matching method with these features (Baseline2).

10 Results

The evaluation results obtained using the baseline methods are given in Table 3.

Method	**Precision**	**Recall**	**Accuracy**	**F_1 score**
Baseline1	0.79	0.40	0.82	0.53
Baseline2	0.95	0.43	0.85	0.60

Table 3. Baseline evaluation results.

The evaluation results obtained from our system by varying the overall hedging score and using it as threshold are given in Table 4. It is worth noting that the highest overall hedging score we obtained was 16; however, we do not show the results for every possible threshold here for brevity.

Hedging Score Threshold	**Precision**	**Recall**	**Accuracy**	**F_1 score**
1	0.68	0.95	0.88	0.79
2	0.75	0.94	0.91	0.83
3	**0.85**	**0.86**	**0.93**	**0.85**
4	0.91	0.71	0.91	0.80
5	0.92	0.63	0.89	0.75
6	0.97	0.40	0.85	0.57
7	1	0.19	0.79	0.33

Table 4. Evaluation results from our system.

As seen from Table 3 and Table 4, our results show improvement over both baseline methods in terms of accuracy and F_1 score. Increasing the threshold (thereby requiring more or stronger hedging devices to qualify a sentence as speculative) improves the precision while lowering the recall. The best accuracy and F_1 score are achieved at threshold t=3. At this threshold, the differences between the results obtained with our method and baseline methods are statistically significant at 0.01 level ($p < 0.01$).

Method	Recall/Precision BEP
Baseline1	0.60
Baseline2	0.76
Our system	0.85

Table 5. Recall / precision break-even point (BEP) results

With the threshold providing the best accuracy and F_1 score, precision and recall are roughly the same (0.85), indicating a recall/precision BEP of approximately 0.85, also an improvement over 0.76 achieved with a weakly supervised classifier (Medlock and Briscoe, 2007). Recall/precision BEP scores are given in Table 5.

11 Discussion

Our results confirm that writers of scientific articles employ basic, predictable hedging strategies to soften their claims or to indicate uncertainty and demonstrate that these strategies can be captured using a combination of lexical and syntactic means. Furthermore, the results indicate that hedging cues can be gainfully weighted to provide a rough measure of tentativeness or uncertainty. For instance, a sentence with the highest overall hedging score is given below:

(7) In one study, Liquid facets was *proposed to* target Dl to an endocytic recycling compartment *suggesting that* recycling of Dl *may* be required for signaling.

On the other hand, hedging is not strong in the following sentence, which is assigned an overall hedging score of 2:

(8) There is no apparent need for cytochrome c release in C. elegans since CED-4 does not require it to activate CED-3.

Below, we discuss some of the common error types we encountered. Our discussion is based on evaluation at hedging score threshold of 0, where existence of a hedging cue is sufficient to label a sentence speculative.

Most of the false negatives produced by the system are due to syntactic patterns not addressed by our method. For instance, negation of "unhedgers" was used as a syntactic pattern; the pattern was able to recognize *know* as an "unhedger" in the following sentence, but not the negative quantifier (*little*), labeling the sentence as non-speculative.

(9) *Little* was *known* however about the specific role of the roX RNAs during the formation of the DCC.

In fact, Hyland (1998) notes "negation in scientific research articles shows a preference for negative quantifiers (*few, little*) and lexical negation (*rarely, overlook*)." However, we have not encountered this pattern while analyzing the training set and have not addressed it. Nevertheless, our approach lends itself to incremental development and adding such a pattern to our rulebase is relatively simple.

Another type of false negative is caused by certain derivational forms of epistemic words. In the following example, the adjective *suggestive* is not recognized as a hedging trigger, even though its base form *suggest* is an epistemic verb.

(10) Phenotypic differences are suggestive of distinct functions for some of these genes in regulating dendrite arborization.

It seems that more sophisticated lexicon expansion rules can be employed to handle such cases. For example, WordNet's "derivationally related form" feature may be used as the basis of these expansion rules.

Regarding false positives, most of them are due to word sense ambiguity concerning hedging cues. For instance, the modal auxiliary *could* is frequently used as a past tense form of *can* in scientific articles to express the role of enabling conditions and external constraints on the occurrence of the proposition rather than uncertainty or tentativeness regarding the proposition. Currently, our system is unable to recognize such cases. An example is given below:

(10) Also we could not find any RAG-like sequences in the recently sequenced sea urchin lancelet hydra and sea anemone genomes, which encode RAG-like sequences.

The context around the hedging cue seems to play a role in these cases. First person plural pronoun (*we*) and/or reference to objective enabling conditions seem to be a common characteristic among false positive cases of *could*.

In other cases, such as *appear*, in the absence of strengthening syntactic cues (*to, that*), we lower the hedging score; however, depending on the threshold, this may not be sufficient to render the sentence non-speculative. Rather than lowering the score equally for all epistemic verbs, a more

appropriate approach would be to consider verb senses separately (e.g., *appear* should be effectively unhedged without a strengthening cue, while *suggest* should only be weakened).

Another type of false positives concern "weak" hedging cues, such as epistemic deductive verbs (*conclude, estimate*) as well as adverbs (*essentially, usually*) and nominalizations (*implication, assumption*).

We have also seen a few instances, which seem speculative on the surface, but were labeled non-speculative. An example is given below:

(11) Caspases can also be activated with the aid of Apaf-1, which in turn *appears to be* regulated by cytochrome c and dATP.

12 Conclusion and Future Work

In this paper, we present preliminary experiments we conducted in recognizing speculative sentences. We draw on previous linguistic work and extend it via semi-automatic methods of lexical acquisition. Using a corpus specifically annotated for speculation, we demonstrate that our linguistically oriented approach improves on the previously reported results.

Our next goal is to extend our work using a larger, more comprehensive corpus. This will allow us to identify other commonly used hedging strategies and refine and expand the hedging dictionary. We also aim to refine the weighting scheme in a more principled way.

While recognizing that a sentence is speculative is useful in and of itself, it seems more interesting and clearly much more challenging to identify speculative sentence fragments and the propositions that are being hedged. In the future, we will move in this direction with the goal of characterizing the semantics of speculative language.

Acknowledgements

We would like to thank Thomas C. Rindflesch for his suggestions and comments on the first draft of this paper.

References

deMarneffe, M. C., MacCartney B., Manning C.D. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. *In Proc of 5th International Conference on Language Resources and Evaluation,* pp. 449-54.

DiMarco C. and Mercer R.E. 2004. Hedging in Scientific Articles as a Means of Classifying Citations. *In Exploring Attitude and Affect in Text: Theories and Applications AAAI-EAAT 2004*. pp.50-4.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA.

Friedman C., Alderson P., Austin J., Cimino J.J., Johnson S.B. 1994. A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2): 161-74.

Hyland K. 1998. *Hedging in Scientific Research Articles*. John Benjamins B.V., Amsterdam, Netherlands.

Kipper Schuler, K. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon.* PhD thesis, University of Pennsylvania.

Klein D. and Manning C. D. 2003. Accurate unlexicalized parsing. *In Proc of 41st Meeting of the Association for Computational Linguistics.* pp. 423-30.

Lakoff G. 1972. Hedges: A Study in Meaning Criteria and the Logic of Fuzzy Concepts. *Chicago Linguistics Society Papers*, 8, pp.183-228.

Light M., Qiu X.Y., Srinivasan P. 2004. The Language of Bioscience: Facts, Speculations, and Statements in between. *In BioLINK 2004: Linking Biological Literature, Ontologies and Databases,* pp. 17-24.

McCray A. T., Srinivasan S., Browne A. C. 1994. Lexical methods for managing variation in biomedical terminologies. *In Proc of 18th Annual Symposium on Computer Applications in Medical Care*, pp. 235-9.

Medlock B. and Briscoe T. 2007. Weakly Supervised Learning for Hedge Classification in Scientific Literature. *In Proc of 45th Meeting of the Association for Computational Linguistics*. pp.992-9.

Palmer F.R. 1986. *Mood and Modality*. Cambridge University Press, Cambridge, UK.

Pustejovsky J., Hanks P., Saurí R., See A., Gaizauskas R., Setzer A., Radev D., Sundheim B., Day D. Ferro L., Lazo M. 2003. The TimeBank Corpus. *In Proc of Corpus Linguistics*. pp. 647-56.

Saurí R., Verhagen M., Pustejovsky J. 2006. SlinkET: a partial modal parser for events. *In Proc of 5th International Conference on Language Resources and Evaluation.*

Wilbur W.J., Rzhetsky A., Shatkay H. 2006. New Directions in Biomedical Text Annotations: Definitions, Guidelines and Corpus Construction. *BMC Bioinformatics*, 7:356.

Cascaded Classifiers for Confidence-Based Chemical Named Entity Recognition

Peter Corbett
Unilever Centre For
Molecular Science Informatics
Chemical Laboratory University Of Cambridge
CB2 1EW, UK
ptc24@cam.ac.uk

Ann Copestake
Computer Laboratory
University Of Cambridge
CB3 0FD, UK
aac10@cl.cam.ac.uk

Abstract

Chemical named entities represent an important facet of biomedical text. We have developed a system to use character-based n-grams, Maximum Entropy Markov Models and rescoring to recognise chemical names and other such entities, and to make confidence estimates for the extracted entities. An adjustable threshold allows the system to be tuned to high precision or high recall. At a threshold set for balanced precision and recall, we were able to extract named entities at an F score of 80.7% from chemistry papers and 83.2% from PubMed abstracts. Furthermore, we were able to achieve 57.6% and 60.3% recall at 95% precision, and 58.9% and 49.1% precision at 90% recall. These results show that chemical named entities can be extracted with good performance, and that the properties of the extraction can be tuned to suit the demands of the task.

1 Introduction

Systems for the recognition of biomedical named entities have traditionally worked on a 'first-best' approach, where all of the entities recognised have equal status, and precision and recall are given roughly equal importance. This does not reflect that fact that precision is of greater importance for some applications, and recall is the key for others. Furthermore, knowing the confidence[1] with which the system has assigned the named entities is likely to be useful in a range of different applications.

Named entities of relevance to biomedical science include not only genes and proteins but also other chemical substances which can be of interest as drugs, metabolites, nutrients, enzyme cofactors, experimental reagents and in many other roles. We have recently investigated the issue of chemical named entities (Corbett et al., 2007), by compiling a set of manual annotation guidelines, demonstrating 93% interannotator agreement and manually annotating a set of 42 chemistry papers. In this paper we demonstrate a named entity recogniser that assigns a confidence score to each named entity, allowing it to be tuned for high precision or recall.

Our review of the methods of chemical named entity recognition showed a consistent theme: the use of character-based n-Grams to identify chemical names via their constituent substrings (Wilbur et al., 1999; Vasserman, 2004; Townsend et al., 2005). This can be a powerful technique, due to systematic and semisystematic chemical names and additional conventions in drug names. However this technique does not cover all aspects of chemical nomenclature.

Much current named entity work uses approaches which combine the structured prediction abilities of HMMs and their derivatives with techniques which enable the use of large, diverse feature sets such as maximum entropy (also known as logistic regression). Maximum Entropy Markov Models, (MEMMs) (McCallum et al., 2000) provide a relatively simple framework for this. MEMMs do have a theoretical weakness, namely the "label bias" problem (Lafferty et al., 2001), which has been ad-

[1] In this paper, we use "confidence" to refer to a system's estimate of the probability that a potential named entity is a correct named entity.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 54–62,
Columbus, Ohio, USA, June 2008.

dressed with the development of Conditional Random Fields (CRFs). CRFs are now a mainstay of the field, being used in a high proportion of entries in the latest BioCreative evaluation (Krallinger and Hirschman, 2007). However, despite the label bias problem, MEMMs still attract interest due to practical advantages such as shorter training cycles.

The framework of HMMs and their successors offers three modes of operation; first-best, n-best and confidence-based. In first-best NER, the Viterbi algorithm is used to identify a single sequence of labels for the target sentence. In n-best operation, the n best sequences for the sentence are identified, along with their probabilities, for example by coupling the Viterbi algorithm with A* search. In confidence-based operation, potential entities (with a probability above a threshold) are identified directly, without directly seeking a single optimal labelling for the entire sentence. This is done by examining the probability of the label transitions within the entity, and the forward and backward probabilities at the start and end of the entity. This mode has been termed the Constrained Forward-Backward algorithm (Culotta and McCallum, 2004). Where a single unambiguous non-overlapping labelling is required, it can be obtained by identifying cases where the entities overlap, and discarding those with lower probabilities.

Confidence-based extraction has two main advantages. First, it enables the balance between precision and recall to be controlled by varying the probability threshold. Second, confidence-based NER avoids over-commitment in systems where it is used as a preprocessor, since multiple overlapping options can be used as input to later components.

The optimum balance between recall and precision depends on the application of the NER and on the other components in the system. High precision is useful in search even when recall is low when there is a large degree of redundancy in the information in the original documents. High precision NER may also be useful in contexts such as the extraction of seed terms for clustering algorithms. Balanced precision/recall is often appropriate for search, although in principle it is desirable to be able to shift the balance if there are too many/too few results. Balanced precision/recall is also generally assumed for use in strictly pipelined systems, when a single set of consistent NER results is to be passed on to subsequent processing. Contexts where high recall is appropriate include those where a search is being carried out where there is little redundancy (cf Carpenter 2007) or where the NER system is being used with other components which can filter the results.

One use of our NER system is within a language processing architecture (Copestake et al., 2006) that systematically allows for ambiguity by treating the input/output of each component as a lattice (represented in terms of standoff annotation on an original XML document). This system exploits relatively deep parsing, which is not fully robust to NER errors but which can exploit complex syntactic information to select between candidate NER results. NER preprocessing is especially important in the context of chemistry terms which utilise punctuation characters (e.g., '2,4-dinitrotoluene', '2,4- and 2,6-dinitrotoluene') since failure to identify these will lead to tokenisation errors in the parser. Such errors frequently cause complete parse failure, or highly inaccurate analyses. In our approach, the NER results contribute edges to a lattice which can (optionally) be treated as tokens by the parser. The NER results may compete with analyses provided by the main parser lexicon. In this context, some NER errors are unimportant: e.g., the parser is not sensitive to all the distinctions between types of named entity. In other cases, the parser will filter the NER results. Hence it makes sense to emphasise recall over precision. We also hypothesise that we will be able to incorporate the NER confidence scores as features in the parse ranking model.

Another example of the use of high-recall NER in an integrated system is shown in the editing workflows used by the Royal Society of Chemistry in their Project Prospect system (Batchelor and Corbett, 2007), where chemical named entity recognition is used to produce semantically-enriched journal articles. In this situation, high recall is desirable, as false positives can be removed in two ways; by removing entities where a chemical structure cannot be assigned, and by having them checked by a technical editor. False negatives are harder to correct.

The use of confidence-based recognition has been demonstrated with CRFs in the domain of contact details (Culotta and McCallum, 2004), and using HMMs in the domain of gene annotation (Carpen-

ter, 2007). In the latter case, the LingPipe toolkit was used in the BioCreative 2 evaluation without significant adaptation. Although only 54% precision was achieved at 60% recall (the best systems were achieving precision and recall scores in the high eighties), the system was capable of 99.99% recall with 7% precision, and 95% recall with 18% precision, indicating that very high recall could be obtained in this difficult domain.

Another potential use of confidence-based NER is the potential to rescore named entities. In this approach, the NER system is run, generating a set of named entities. Information obtained about these entities throughout the document (or corpus) that they occur in can then be used in further classifiers. We are not aware of examples of rescoring being applied to confidence-based NER, but there are precedents using other modes of operations. For example, Krishnan and Manning (2006) describe a system where a first-best CRF is used to analyse a corpus, the results of which are then used to generate additional features to use in a second first-best CRF. Similarly, Yoshida and Tsujii (2007) use an n-best MEMM to generate multiple analyses for a sentence, and re-rank the analyses based on information extracted from neighbouring sentences.

Therefore, to explore the potential of these techniques, we have produced a chemical NER system that uses a MEMM for confidence-based extraction of named entities, with an emphasis on the use of character-level n-Grams, and a rescoring system.

2 Corpus

Previously, we have produced a set of annotation guidelines for chemical named entities, and used them to annotate a set of 42 chemistry papers (Corbett et al., 2007). Inter-annotator agreement was tested on 14 of these, and found to be 93%. The annotation guidelines specified five classes of named entity, which are detailed in Table 1. The annotation was performed on untokenised text.

To test the applicability of the method to a different corpus, we retrieved 500 PubMed abstracts and titles, and annotated them using the same methods. The abstracts were acquired using the query `metabolism[Mesh] AND drug AND hasabstract`. This produced a diverse set of abstracts spanning a wide range of subject areas, but which contain a higher proportion of relevant terms than PubMed overall. 445 out of 500 abstracts contained at least one named entity, whereas 249 contained at least ten. Notably, the `ASE` class was more common in the PubMed corpus than in the chemistry papers, reflecting the important of enzymes to biological and medical topics.

In this study, we have left out the named entity type `CPR`, as it is rare (<1%) and causes difficulties with tokenisation. This entity type covers cases such as the "1,3-" in "1,3-disubstituted", and as such requires the "1,3-" to be a separate token or token sequence. However, we have found that recognition of the other four classes is improved if words such as "1,3-disubstituted" are kept together as single tokens. Therefore it makes sense to treat the recognition of `CPR` as an essentially separate problem - a problem that will not be addressed here.

Type	Description	Example	n_{Ch}	n_{PM}
CM	compound	citric acid	6865	4494
RN	reaction	methylation	288	401
CJ	adjective	pyrazolic	60	87
ASE	enzyme	demethylase	31	181
CPR	prefix	1,3-	53	21

Table 1: Named Entity types. n_{Ch} = number in Chemistry corpus, n_{PM} = number in PubMed corpus.

3 Methods

Our system is quite complex, and as such we have made the source code available (see below). The following gives an outline of the system:

3.1 External Resources

Chemical names were extracted from the chemical ontology ChEBI (Degtyarenko et al., 2008), and a standard English word list was taken from `/usr/share/dict/words` on a Linux system[2]. A list of chemical element names and symbols was also compiled. To overcome the shortage of entities of type `ASE`, a list of words from enzyme names

[2]This dictionary was chosen as it contains inflectional forms of English words. Our system does not perform stemming, partly because suffixes are often good cues as to whether a word is chemical or not.

ending in '-ase' was extracted from the Gene Ontology (GO), and hand sorted into words of type ASE, and words not of type ASE.

3.2 Overview of operation

The text is tokenised before processing; this is done using the tokeniser described in our previous work (Corbett et al., 2007), which is adapted to chemical text.

Our system uses three groups of classifiers to recognise chemical names. The first classifier—the 'preclassifier'—uses character-level n-grams to estimate the probabilities of whether tokens are chemical or not. The output of this classification is combined with information from the suffix of the word, and is used to provide features for the MEMM.

The second group of classifiers constitute the MEMM proper. Named entities are represented using an BIO-encoding, and methods analogous to other confidence-based taggers (Culotta and McCallum, 2004; Carpenter, 2007) are used to estimate the conditional probability of tag sequences corresponding to named entities. The result of this is a list of potential named entities, with start positions, end positions, types and probabilities, where all of the probabilities are above a threshold value. A small set of hand-written filtering rules is used to remove obvious absurdities, such as named entities ending in the word "the", and simple violations of the annotation guidelines, such as named entities of type ASE that contain whitespace. These filtering rules make very little difference at recall values up to about 80%—however, we have found that they are useful for improving precision at very high recall.

The third group of classifiers—one per entity type—implement a rescoring system. After all of the potential entities from a document have been generated, a set of features is generated for each entity. These features are derived from the probabilities of other entities that share the same text string as the entity, from probabilities of potential synonyms found via acronym matching and other processes, and most importantly, from the pre-rescoring probability of the entities themselves. In essence, the rescoring process performs Bayesian reasoning by adjusting the raw probabilities from the previous stage up or down based on nonlocal information within the document.

3.3 Overview of training

A form of training conceptually similar to cross-validation is used to train the three layers of classifiers. To train the overall system, the set of documents used for training is split into three. Two thirds are used to train a MEMM, which is then used to generate training data for the rescorer using the held-out last third. This process is repeated another two times, holding out a different third of the training data each time. Finally, the rescorer is trained using all of the training data generated by this procedure, and the final version of the MEMM is generated using all of the training data. This procedure ensures that both the MEMM and the rescorer are able to make use of all of the training data, and also that the rescorer is trained to work with the output of a MEMM that has not been trained on the documents that it is to rescore.

A similar procedure is used when training the MEMM itself. The available set of documents to use as training data is divided into half. One half is used to train the preclassifier and build its associated dictionaries, which are then used to generate features for the MEMM on the other half of the data. The roles of each half are then reversed, and the same process is applied. Finally, the MEMM is trained using all of the generated features, and a new preclassifier is trained using all of the available training data.

It should be noted that the dictionaries extracted during the training of the preclassifier are also used directly in the MEMM.

3.4 The character n-gram based preclassifier

During the training of the preclassifier, sets of tokens are extracted from the hand-annotated training data. A heuristic is used to classify these into 'word tokens'—those that match the regex `.*[a-z][a-z].*`, and 'nonword tokens'—those that do not (this class includes many acronyms and chemical formulae). The n-gram analysis is only performed upon 'word tokens'.

The token sets that are compiled are chemical word tokens (those that only appear inside named entities), nonchemical word tokens (those that do not appear in entities), chemical nonword tokens, nonchemical nonword tokens and ambigu-

ous tokens—those that occur both inside and outside of named entities. A few other minor sets are collected to deal with tokens related to such proper noun-containing entities as 'Diels–Alder reaction'.

Some of this data is combined with external dictionaries to train the preclassifier, which works using 4-grams of characters and modified Kneser-Ney smoothing, as described by Townsend et al. (2005). The set of 'chemical word tokens' is used as a set of positive examples, along with tokens extracted from ChEBI, a list of element names and symbols, and the `ASE` tokens extracted from the GO. The negative examples used are the extracted 'nonchemical word tokens', the non-`ASE` tokens from the GO and tokens taken from the English dictionary—except for those that were listed as positive examples. This gets around the problem that the English dictionary contains the names of all of the elements and a number of simple compounds such as 'ethanol'.

During operation, n-gram analysis is used to calculate a score for each word token, of the form:

$$ln(P(token|chem)) - ln(P(token|nonchem))$$

If this score is above zero, the preclassifier classifies the token as chemical and gives it a tentative type, based on its suffix. This can be considered to be a "first draft" of its named entity type. For example tokens ending in "-ation" are given the type `RN`, whereas those ending in "-ene" are given type `CM`.

3.5 The MEMM

The MEMM is a first-order MEMM, in that it has a separate maximum-entropy model for each possible preceeding tag. No information about the tag sequence was included directly in the feature set. We use the OpenNLP MaxEnt classifier[3] for maximum-entropy classification.

The feature set for the MEMM is divided into three types of features; type 1 (which apply to the token itself), type 2 (which can apply to the token itself, the previous token and the next token) and type 3 (which can act as type 2 features, and which can also form bigrams with other type 3 features).

An example type 1 feature would be `4G=ceti`, indicating that the 4-gram `ceti` had been found in the token. An example type 2 feature would be `c-1:w=in`, indicating that the previous token was 'in'. An example bigram constructed from type 3 features would be `bg:0:1:ct=CJ__w=acid`, indicating that the preclassifier had classified the token as being of type `CJ`, and having a score above zero, and that the next token was 'acid'.

Type 1 features include 1, 2, 3 and 4-grams of characters found within the token, whether the token appeared in any of the word lists, and features to represent the probability and type given by the preclassifier for that token. Type 2 features include the token itself with any terminal letter 's' removed, the token converted to lowercase (if it matched the regex `.*[a-z][a-z].*`), and a three-character suffix taken from the token. The token itself was usually used as a type 2 feature, unless it unless it was short (less than four characters), or had been found to be an ambiguous token during preclassifier training, in which case it was type 3. Other type 3 features include a word shape feature, and tentative type of the token if the preclassifier had classed it as chemical.

A few other features were used to cover a few special cases, and were found to yield a slight improvement during development.

After generating the features, a feature selection based on log-likelihood ratios is used to remove the least informative features, with a threshold set to remove about half of them. This was found during development to have only a very small beneficial effect on the performance of the classifier, but it did make training faster and produced smaller models. This largely removed rare features which were only found on a few non-chemical tokens.

3.6 The rescorer

The rescoring system works by constructing four maximum entropy classifiers, one for each entity type. The output of these classifiers is a probability of whether or not a potential named entity really is a correct named entity. The generation of features is done on a per-document basis.

The key features in the rescorer represent the probability of the potential entity as estimated by the MEMM. The raw probability p is converted to the logit score

$$l = ln(p) - ln(1 - p)$$

This mirrors the way probabilities are represented

[3] `http://maxent.sourceforge.net/`

within maximum entropy (*aka* logistic regression) classifiers. If l is positive, $int(min(15.0, l) * 50)$ instances [4] of the feature `conf+` are generated, and a corresponding technique is used if l is negative.

Before generating further features, it is necessary to find entities that are 'blocked'—entities that overlap with other entities of higher confidence. For example, consider "ethyl acetate", which might give rise to the named entity "ethyl acetate" with 98% confidence, and also "ethyl" with 1% confidence and "acetate" with 1% confidence. In this case, "ethyl" and "acetate" would be blocked by "ethyl acetate".

Further features are generated by collecting together all of the unblocked[5] potential entities of a type that share the same string, calculating the maximum and average probability, and calculating the difference between the p and those quantities.

Some acronym and abbreviation handling is also performed. The system looks for named entities that are surrounded by brackets. For each of these, a list of features is generated that is then given to every other entity of the same string. If there is a potential entity to the left of the bracketed potential abbreviation, then features are generated to represent the probability of that potential entity, and how well the string form of that entity matches the potential abbreviation. If no potential entity is found to match with, then features are generated to represent how well the potential abbreviation matches the tokens to the left of it. By this method, the rescorer can gather information about whether a potential abbreviation stands for a named entity, something other than a named entity—or whether it is not an abbreviation at all, and use that information to help score all occurrences of that abbreviation in the document.

4 Evaluation

The systems were evaluated by 3-fold cross-validation methodology, whereby the data was split into three equal folds (in the case of the chemistry papers, each fold consists of one paper per journal. For the PubMed abstracts, each fold consists of one third of the total abstracts). For each fold, the system was trained on the other two folds and then evaluated on that fold, and the results were pooled.

The direct output from the system is a list of putative named entities with start positions, end positions, types and confidence scores. This list was sorted in order of confidence—most confident first—and each entity was classified as a true positive or a false positive according to whether an exact match (start position, end position and type all matched perfectly) could be found in the annotated corpus. Also, the number of entities in the annotated corpus was recorded.

Precision/recall curves were plotted from these lists by selecting the first n elements, and calculating precision and recall taking all of the elements in this sublist as true or false positives, and all the entities in the corpus that were not in the sublist as false negatives. The value of n was gradually increased, recording the scores at each point. The area under the curve (treating precision as zero at recall values higher than the highest reported) was used to calculate mean average precision (MAP). Finally, F were generated by selecting all of the entities with a confidence score of 0.3 or higher.

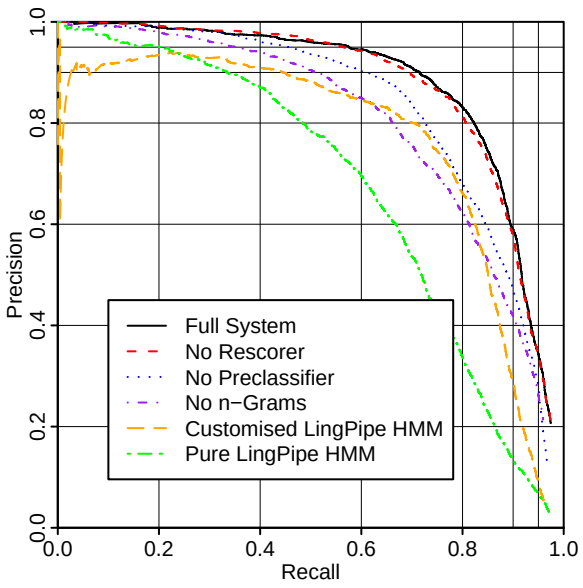

Figure 1: Evaluation on chemistry papers.

The results of this evaluation on the corpus of

[4] We found that 15.0 was a good threshold by experimentation on development data: papers annotated during trial runs of the annotation process.

[5] Doing this without regards for blocking causes problems. In a document containing both "ethyl acetate" and "ethyl group", it would be detrimental to allow the low confidence for the "ethyl" in "ethyl acetate" to lower the confidence of the "ethyl" in "ethyl group".

chemistry papers is show in Figure 1. The full system achieves 57.6% recall at 95% precision, 58.9% precision at 90% recall, and 78.7% precision and 82.9% recall (F = 80.7%) at a confidence threshold of 0.3. Also shown are the results of successively eliminating parts of the system. "No Rescorer" removes the rescorer. In "No Preclassifier", the preclassifier is disabled, and all of the dictionaries extracted during the training of the preclassifier are also disabled. Finally, in "No n-Grams", the 1-, 2-, 3- and 4-grams used directly by the MEMM are also disabled, showing the results of using a system where no character-level n-grams are used at all. These modifications apply successively—for example, in the "No n-Grams" case the rescorer and preclassifier are also disabled. These results validate the the cascade of classifiers, and underline the importance of character-level n-grams in chemical NER.

We also show comparisons to an HMM-based approach, based on LingPipe 3.4.0.[6] This is essentially the same system as described by Corbett et al. (2007), but operating in a confidence-based mode. The HMMs used make use of character-level n-Grams, but do not allow the use of the rich feature set used by the MEMM. The line "Customised LingPipe HMM" shows the system using the custom tokenisation and ChEBI-derived dictionary used in the MEMM system, whereas the "Pure LingPipe HMM" shows the system used with the default tokeniser and no external dictionaries. In the region where precision is roughly equal to recall (mimicking the operation of a first-best system), the fact that the MEMM-based system outperforms an HMM is no surprise. However, it is gratifying that a clear advantage can be seen throughout the whole recall range studied (0-97%), indicating that the training processes for the MEMM are not excessively attuned to the first-best decision boundary. This increased accuracy comes at a price in the speed of development, training and execution.

It is notable that we were not able to achieve extremes of recall at tolerable levels of precision using any of the systems, whereas it was possible for LingPipe to achieve 99.99% recall at 7% precision in the BioCreative 2006 evaluation. There are a number of potential reasons for this. The first is that the tokeniser used in all systems apart from the "Pure LingPipe HMM" system tries in general to make as few token boundaries as possible; this leads to some cases where the boundaries of the entities to be recognised in the test paper occur in the middle of tokens, thus making those entities unrecognisable whatever the threshold. However this does not appear to be the whole problem. Other factors that may have had an influence include the more generous method of evaluation at BioCreative 2006, (where several allowable alternatives were given for difficult named entities), and the greater quantity and diversity (sentences selected from a large number of different texts, rather than a relatively small number of whole full papers) of training data. Finally, there might be some important difference between chemical names and gene names.

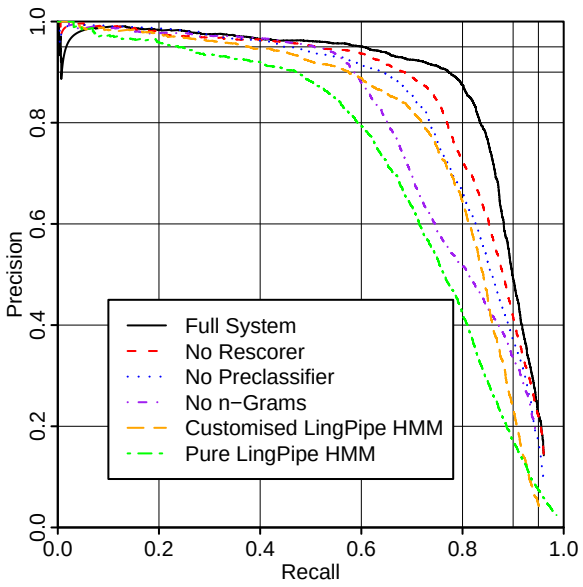

Figure 2: Evaluation on PubMed abstracts.

Figure 2 shows the results of running the system on the set of annotated PubMed abstracts. The full system achieves 60.3% recall at 95% precision, 49.1% precision at 90% recall, and 85.0% precision and 81.6% recall (F = 83.2%) at a confidence threshold of 0.3. In PubMed abstracts, it is common to define ad-hoc abbreviations for chemicals within an abstract (e.g., the abstract might say 'dexamethasone (DEX)', and then use 'DEX' and not 'dexamethasone' throughout the rest of the abstract). The rescorer provides a good place to resolve these ab-

[6] http://alias-i.com/lingpipe/

breviations, and thus has a much larger effect than in the case of chemistry papers where these ad hoc abbreviations are less common. It is also notable that the maximum recall is lower in this case. One system—the "Pure LingPipe HMM", which uses a different, more aggressive tokeniser from the other systems—has a clear advantage in terms of maximum recall, showing that overcautious tokenisation limits the recall of the other systems.

In some cases the systems studied behave strangely, having "spikes" of lowered precision at very low recall, indicating that the systems can occasionally be overconfident, and assign very high confidence scores to incorrect named entities.

Corpus	System	MAP	F
Chemistry	Full	87.1%	80.8%
Chemistry	No Rescorer	86.8%	81.0%
Chemistry	No Preclassifier	82.7%	74.8%
Chemistry	No n-Grams	79.2%	72.2%
Chemistry	Custom LingPipe	75.9%	74.6%
Chemistry	Pure LingPipe	66.9%	63.2%
Chemistry	No Overlaps	82.9%	80.8%
PubMed	Full	86.1%	83.2%
PubMed	No Rescorer	83.3%	79.1%
PubMed	No Preclassifier	81.4%	73.4%
PubMed	No n-Grams	77.6%	70.6%
PubMed	Custom LingPipe	78.6%	75.6%
PubMed	Pure LingPipe	71.9%	66.1%

Table 2: F scores (at confidence threshold of 0.3) and Mean Average Precision (MAP) values for Figs. 1-3.

Neither corpus contains enough data for the results to reach a plateau—using additional training data is likely to give improvements in performance.

The "No Overlaps" line in Figure 3 shows the effect of removing "blocked" named entities (as defined in section 3.6) prior to rescoring. This simulates a situation where an unambiguous inline annotation is required—for example a situation where a paper is displayed with the named entities being highlighted. This condition makes little difference at low to medium recall, but it sets an effective maximum recall of 90%. The remaining 10% of cases presumably consist of situations where the recogniser is finding an entity in the right part of the text, but making boundary or type errors.

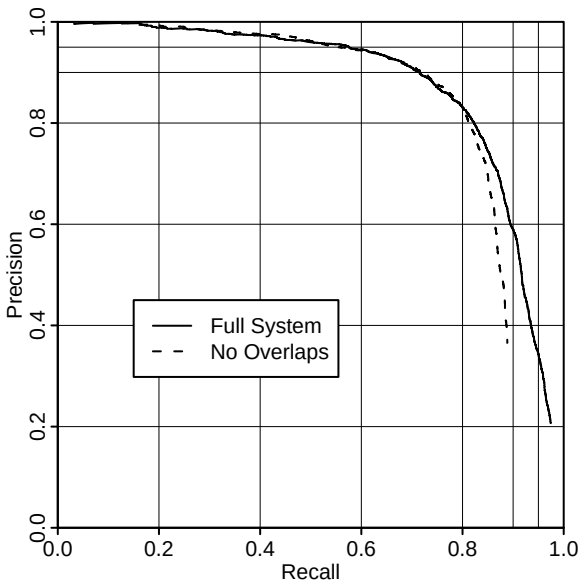

Figure 3: Evaluation on chemistry papers, showing effects of disallowing overlapping entities.

5 Conclusion

We have demonstrated that MEMMs can be adapted to recognise chemical named entities, and that the balance between precision and recall can be tuned effectively, at least in the range of 0 - 95% recall. The MEMM system is available as part of the OSCAR3 chemical named entity recognition system. [7]

Acknowledgements

PTC thanks Peter Murray-Rust for supervision. We thank the Royal Society of Chemistry for providing the papers, and the EPSRC (EP/C010035/1) for funding. We thank the reviewers for their helpful suggestions and regret that we did not have the time or space to address all of the issues raised.

References

Colin Batchelor and Peter Corbett. 2007. Semantic enrichment of journal articles using chemical named entity recognition *Proceedings of the ACL 2007 Demo and Poster Sessions*, pp 45-48. Prague, Czech Republic.

Bob Carpenter. 2007. LingPipe for 99.99% Recall of Gene Mentions *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, 307-309.

[7] https://sourceforge.net/projects/oscar3-chem

Ann Copestake, Peter Corbett, Peter Murray-Rust, C. J. Rupp, Advaith Siddharthan, Simone Teufel and Ben Waldron. 2006. An Architecture for Language Technology for Processing Scientific Texts. *Proceedings of the 4th UK E-Science All Hands Meeting*, Nottingham, UK.

Peter Corbett, Colin Batchelor and Simone Teufel. 2007. Annotation of Chemical Named Entities *BioNLP 2007: Biological, translational, and clinical language processing*, pp 57-64. Prague, Czech Republic.

Aron Culotta and Andrew McCallum 2004. Confidence Estimation for Information Extraction *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pp 109-112. Boston, MA.

Kirill Degtyarenko, Paula de Matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan McNaught, Rafael Alcantara, Michael Darsow, Mickael Guedj and Michael Ashburner. 2008. ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res*, Vol. 36, Database issue D344-D350.

The Gene Ontology Consortium 2000. Gene Ontology: tool for the unification of biology. *Nature Genetics*, Vol. 25, 26-29.

Martin Krallinger and Lynette Hirschman, editors. 2007. *Proceedings of the Second BioCreative Challenge Evaluation Workshop*.

Vijay Krishnan and Christopher D. Manning. 2006. An Effective Two-Stage Model for Exploiting Non-Local Dependencies in Named Entity Recognition. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 1121-1128. Sindey, Australia.

John Lafferty, Andrew McCallum and Fernando Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the Eighteenth International Conference on Machine Learning*, 282-289.

Andrew McCallum, Dayne Freitag and Fernando Pereira. 2000. Maximum Entropy Markov Models for Information Extraction and Segmentation *Proceedings of the Seventeenth International Conference on Machine Learning*, 591-598. San Fransisco, CA.

Joe A. Townsend, Ann Copestake, Peter Murray-Rust, Simone H. Teufel and Christopher A. Waudby. 2005. Language Technology for Processing Chemistry Publications *Proceedings of the fourth UK e-Science All Hands Meeting*, 247-253. Nottingham, UK.

Alexander Vasserman 2004 Identifying Chemical Names in Biomedial Text: An Investigation of the Substring Co-occurence Based Approaches *Proceedings of the Student Research Workshop at HLT-NAACL*

W. John Wilbur, George F. Hazard, Jr., Guy Divita, James G. Mork, Alan R. Aronson and Allen C. Browne. 1999 Analysis of Biomedical Text for Chemical Names: A Comparison of Three Methods *Proc. AMIA Symp.* 176-180.

Kazuhiro Yoshida and Jun'ichi Tsujii. 2007. Reranking for Biomedical Named-Entity Recognition *BioNLP 2007: Biological, translational, and clinical language processing*, pp 57-64. Prague, Czech Republic.

How to Make the Most of NE Dictionaries in Statistical NER

Yutaka Sasaki[2] **Yoshimasa Tsuruoka**[2] **John McNaught**[1,2] **Sophia Ananiadou**[1,2]
[1] National Centre for Text Mining
[2] School of Computer Science, University of Manchester
MIB, 131 Princess Street, Manchester, M1 7DN, UK

Abstract

When term ambiguity and variability are very high, dictionary-based *Named Entity Recognition (NER)* is not an ideal solution even though large-scale terminological resources are available. Many researches on statistical NER have tried to cope with these problems. However, it is not straightforward how to exploit existing and additional *Named Entity (NE)* dictionaries in statistical NER. Presumably, addition of NEs to an NE dictionary leads to better performance. However, in reality, the retraining of NER models is required to achieve this. We have established a novel way to improve the NER performance by addition of NEs to an NE dictionary without retraining. We chose protein name recognition as a case study because it most suffers the problems related to heavy term variation and ambiguity. In our approach, first, known NEs are identified in parallel with *Part-of-Speech (POS)* tagging based on a general word dictionary and an NE dictionary. Then, statistical NER is trained on the *tagger outputs* with correct NE labels attached. We evaluated performance of our NER on the standard JNLPBA-2004 data set. The F-score on the test set has been improved from 73.14 to 73.78 after adding the protein names appearing in the training data to the POS tagger dictionary without any model retraining. The performance further increased to 78.72 after enriching the tagging dictionary with test set protein names. Our approach has demonstrated high performance in protein name recognition, which indicates how to make the most of known NEs in statistical NER.

1 Introduction

The accumulation of online biomedical information has been growing at a rapid pace, mainly attributed to a rapid growth of a wide range of repositories of biomedical data and literature. The automatic construction and update of scientific *knowledge bases* is a major research topic in Bioinformatics. One way of populating these knowledge bases is through *named entity recognition (NER)*. Unfortunately, biomedical NER faces many problems, e.g., protein names are extremely difficult to recognize due to ambiguity, complexity and variability. A further problem in protein name recognition arises at the tokenization stage. Some protein names include punctuation or special symbols, which may cause tokenization to lose some word concatenation information in the original sentence. For example, `IL-2` and `IL - 2` fall into the same token sequence `IL - 2` as usually dash (or hyphen) is designated as a token delimiter.

Research into NER is centred around three approaches: dictionary-based, rule-based and machine learning-based approaches. To overcome the usual NER pitfalls, we have opted for a hybrid approach combining dictionary-based and machine learning approaches, which we call *dictionary-based statistical NER approach*. After identifying protein names in text, we link these to semantic identifiers, such as UniProt accession numbers. In this paper, we focus on the evaluation of our dictionary-based statistical NER.

2 Methods

Our dictionary-based statistical approach consists of two components: dictionary-based POS/PROTEIN tagging and statistical sequential labelling. First,

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 63–70,
Columbus, Ohio, USA, June 2008.

dictionary-based POS/PROTEIN tagging finds candidates for protein names using a dictionary. The dictionary maps strings to parts of speech (POS), where the POS tagset is augmented with a tag NN-PROTEIN. Then, sequential labelling applies to reduce false positives and false negatives in the POS/PROTEIN tagging results. Expandability is supported through allowing a user of the NER tool to improve NER coverage by adding entries to the dictionary. In our approach, retraining is not required after dictionary enrichment.

Recently, *Conditional Random Fields (CRFs)* have been successfully applied to sequence labelling problems, such as POS tagging and NER, and have outperformed other machine learning techniques. The main idea of CRFs is to estimate a conditional probability distribution over label sequences, rather than over local directed label sequences as with Hidden Markov Models (Baum and Petrie, 1966) and Maximum Entropy Markov Models (McCallum et al., 2000). Parameters of CRFs can be efficiently estimated through the log-likelihood parameter estimation using the forward-backward algorithm, a dynamic programming method.

2.1 Training and test data

Experiments were conducted using the training and test sets of the JNLPBA-2004 data set(Kim et al., 2004).

Training data The training data set used in JNLPBA-2004 is a set of tokenized sentences with manually annotated term class labels. The sentences are taken from the Genia corpus (version 3.02) (Kim et al., 2003), in which 2,000 abstracts were manually annotated by a biologist, drawing on a set of POS tags and 36 biomedical term classes. In the JNLPBA-2004 shared task, performance in extracting five term classes, i.e., protein, DNA, RNA, cell line, and cell type classes, were evaluated.

Test Data The test data set used in JNLPBA-2004 is a set of tokenized sentences extracted from 404 separately collected MEDLINE abstracts, where the term class labels were manually assigned, following the annotation specification of the Genia corpus.

2.2 Overview of dictionary-based statistical NER

Figure 1 shows the block diagram of dictionary-based statistical NER. Raw text is analyzed by a POS/PROTEIN tagger based on a CRF tagging model and dictionary, and then converted into token sequences. Strings in the text that match with protein names in the dictionary will be tagged as NN-PROTEIN depending on the context around the protein names. Since it is not realistic to enumerate all protein names in the dictionary, due to their high variability of form, instead previously unseen forms are predicted to be protein names by statistical sequential labelling. Finally, protein names are identified from the POS/PROTEIN tagged token sequences via a CRF labelling model.

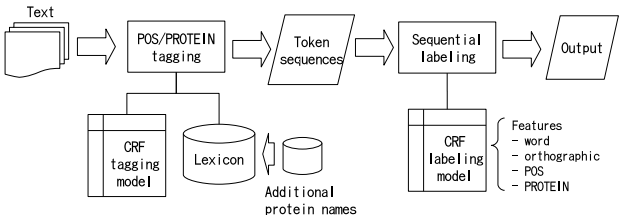

Figure 1: Block diagram of dictionary-based statistical NER

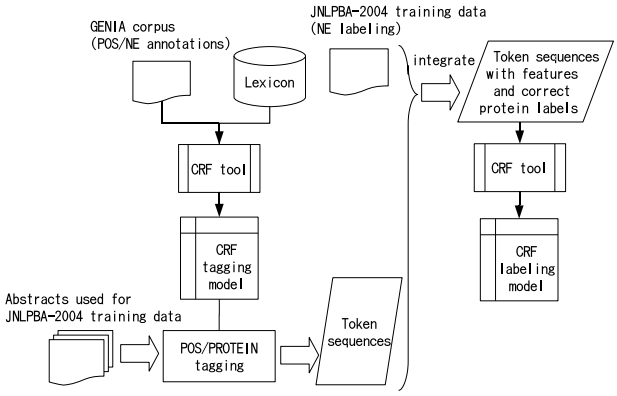

Figure 2: Block diagram of training procedure

Figure 2 shows the block diagram of the training procedure for both POS/PROTEIN tagging and sequential labelling. The tagging model is created using the Genia corpus (version 3.02) and a dictionary. Using the tagging model, MEDLINE abstracts used for the JNLPBA-2004 training data set are then POS/PROTEIN-tagged. The output token sequences over these abstracts are then integrated with the correct protein labels of the JNLPBA-2004 training data. This process results in the preparation of token sequences with features and correct protein labels. A CRF labelling model is finally generated by applying a CRF tool to these decorated token sequences.

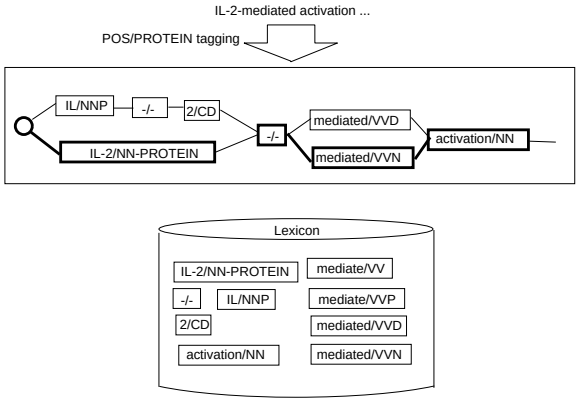

Figure 3: Dictionary based approach

2.2.1 Dictionary-based POS/PROTEIN tagging

The dictionary-based approach is beneficial when a sentence contains some protein names that conflict with general English words. Otherwise, if the POS tags of sentences are decided without considering possible occurrences of protein names, POS sequences could be disrupted. For example, in "met proto-oncogene precursor", *met* might be falsely recognized as a verb by a non dictionary-based tagger.

Given a sentence, the dictionary-based approach extracts protein names as follows. Find all word sequences that match the lexical entries, and create a token graph (i.e., *trellis*) according to the word order. Estimate the score of every path using the weights of node and edges estimated by training using Conditional Random Fields. Select the best path.

Figure 3 shows an example of our dictionary-based approach. Suppose that the input is "IL-2-mediated activation". A trellis is created based on the lexical entries in a dictionary. The selection criteria for the best path are determined by the CRF tagging model trained on the Genia corpus. In this example, `IL-2/NN-PROTEIN -/- mediated/VVN activation/NN` is selected as the best path. Following Kudo et al. (Kudo et al., 2004), we adapted the core engine of the CRF-based morphological analyzer, MeCab[1], to our POS/PROTEIN tagging task. MeCab's dictionary databases employ double arrays (Aoe, 1989) which enable efficient lexical look-ups.

The features used were:

- POS
- PROTEIN
- POS-PROTEIN
- bigram of adjacent POS
- bigram of adjacent PROTEIN
- bigram of adjacent POS-PROTEIN

During the construction of the trellis, white space is considered as the delimiter unless otherwise stated within dictionary entries. This means that unknown tokens are character sequences without spaces.

2.2.2 Dictionary construction

A dictionary-based approach requires the dictionary to cover not only a wide variety of biomedical terms but also entries with:

- all possible capitalization
- all possible linguistic inflections

We constructed a freely available, wide-coverage English word dictionary that satisfies these conditions. We did consider the MedPost pos-tagger package[2] which contains a free dictionary that has downcased English words; however, this dictionary is not well curated as a dictionary and the number of entries is limited to only 100,000, including inflections.

Therefore, we started by constructing an English word dictionary. Eventually, we created a dictionary with about 266,000 entries for English words (systematically covering inflections) and about 1.3 million entries for protein names.

We created the general English part of the dictionary from WordNet by semi-automatically adding POS tags. The POS tag set is a minor modification of the Penn Treebank POS tag set[3], in that protein names are given a new POS tag, NN-PROTEIN. Further details on construction of the dictionary now follow.

Protein names were extracted from the BioThesaurus[4]. After selecting only those terms clearly stated as protein names, 1,341,992 protein names in total were added to the dictionary.

[1] http://sourceforge.net/project/showfiles.php?group_id=177856/`biothesaurus/`

[2] `ftp://ftp.ncbi.nlm.nih.gov/pub/lsmith/MedPost/`

[3] `ftp://ftp.cis.upenn.edu/pub/treebank/doc/tagguide.ps.gz`

[4] `http://pir.georgetown.edu/iprolink/`

Nouns were extracted from WordNet's noun list. Words starting with lower case and upper case letters were determined as NN and NNP, respectively. Nouns in NNS and NNPS categories were collected from the results of POS tagging articles from Plos Biology Journal[5] with TreeTagger[6].

Verbs were extracted from WordNet's verb list. We manually curated VBD, VBN, VBG and VBZ verbs with irregular inflections based on WordNet. Next, VBN, VBD, VBG and VBZ forms of regular verbs were automatically generated from the WordNet verb list.

Adjectives were extracted from WordNet's adjective list. We manually curated JJ, JJR and JJS of irregular inflections of adjectives based on the WordNet irregular adjective list. Base form (JJ) and regular inflections (JJR, JJS) of adjectives were also created based on the list of adjectives.

Adverbs were extracted from WordNet's adverb list. Both the original and capitalised forms were added as RB.

Pronouns were manually curated. PRP and PRP$ words were added to the dictionary.

Wh-words were manually curated. As a result, WDT, WP, WP$ and WRB words were added to the dictionary.

Words for other parts of speech were manually curated.

2.2.3 Statistical prediction of protein names

Statistical sequential labelling was employed to improve the coverage of protein name recognition and to remove false positives resulting from the previous stage (dictionary-based tagging).

We used the JNLPBA-2004 training data, which is a set of tokenized word sequences with IOB2(Tjong Kim Sang and Veenstra, 1999) protein labels. As shown in Figure 2, POSs of tokens resulting from tagging and tokens of the JNLPBA-2004 data set are integrated to yield training data for sequential labelling. During integration, when the single token of a protein name found after tagging corresponds to a sequence of tokens from JNLPBA-2004, its POS is given as NN-PROTEIN1, NN-PROTEIN2,..., according to the corresponding token order in the JNLPBA-2004 sequence.

Following the data format of the JNLPBA-2004 training set, our training and test data use the IOB2 labels, which are "B-protein" for the first token of the target sequence, "I-protein" for each remaining token in the target sequence, and "O" for other tokens. For example, "Activation of the IL 2 precursor provides" is analyzed by the POS/PROTEIN tagger as follows.

```
Activation      NN
of              IN
the             DT
IL 2 precursor  NN-PROTEIN
provides        VVZ
```

The tagger output is given IOB2 labels as follows.

```
Activation NN          O
of         IN          O
the        DT          O
IL         NN-PROTEIN1 B-protein
2          NN-PROTEIN2 I-protein
precursor  NN-PROTEIN3 I-protein
provides   VVZ         O
```

We used CRF models to predict the IOB2 labels. The following features were used in our experiments.

- word feature
- orthographic features
  - the first letter and the last four letters of the word form, in which capital letters in a word are normalized to "A", lower case letters are normalized to "a", and digits are replaced by "0", *e.g.*, the word form of IL-2 is AA-0.
  - postfixes, the last two and four letters
- POS feature
- PROTEIN feature

The window size was set to ± 2 of the current token.

3 Results and discussion

[5] http://biology.plosjournals.org/

[6] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html/

Table 1: Experimental Rusults

Tagging		R	P	F
(a) POS/PROTEIN tagging	Full	52.91	43.85	47.96
	Left	61.48	50.95	55.72
	Right	61.38	50.87	55.63
Sequential Labelling		**R**	**P**	**F**
(b) Word feature	Full	63.23	70.39	66.62
	Left	68.15	75.86	71.80
	Right	69.88	77.79	73.63
(c) (b) + orthographic feature	Full	77.17	67.52	72.02
	Left	82.51	72.20	77.01
	Right	84.29	73.75	78.67
(d) (c) + POS feature	Full	76.46	68.41	72.21
	Left	81.94	73.32	77.39
	Right	83.54	74.75	78.90
(e) (d) + PROTEIN feature	Full	77.58	69.18	73.14
	Left	82.69	73.74	77.96
	Right	84.37	75.24	79.54
(f) (e) + after adding protein names in the training set to the dictionary	Full	**79.85**	**68.58**	**73.78**
	Left	84.82	72.85	78.38
	Right	86.60	74.37	80.02

3.1 Protein name recognition performance

Table 1 shows our protein name recognition results, showing the differential effect of various combinations of strategies. Results are expressed according to recall (R), precision (P), and F-measure (F), which here measure how accurately our various experiments determined the left boundary (Left), the right boundary (Right), and both boundaries (Full) of protein names. The baseline for tagging (row (a)) shows the protein name detection performance of our dictionary-based tagging using our large protein name dictionary, where no training for protein name prediction was involved. The F-score of this baseline tagging method was 47.96.

The baseline for sequential labelling (row (b)) shows the prediction performance when using only word features where no orthographic and POS features were used. The F-score of the baseline labelling method was 66.62. When orthographic feature was added (row (c)), the F-score increased by 5.40 to 72.02. When the POS feature was added (row (d)), the F-score increased by 0.19 to 72.21. Using all features (row (e)), the F-score reached 73.14. Surprisingly, adding protein names appearing in the *training data* to the dictionary further improved the F-score by 0.64 to 73.78, which is the second best score for protein name recognition using the JNLPBA-2004 data set.

Table 2: After Dictionary Enrichment

Method		R	P	F
Tagging (+test set protein names)	Full	79.02	61.87	69.40
	Left	82.28	64.42	72.26
	Right	80.96	63.38	71.10
Labelling (+test set protein names)	full	**86.13**	**72.49**	**78.72**
	Left	89.58	75.40	81.88
	Right	90.23	75.95	82.47

Tagging and labelling speeds were measured using an unloaded Linux server with quad 1.8 GHz Opteron cores and 16GB memory. The dictionary-based POS/PROTEIN tagger is very fast even though the total size of the dictionary is more than one million. The processing speed for tagging and sequential labelling of the 4,259 sentences of the test set data took 0.3 sec and 7.3 sec, respectively, which means that in total it took 7.6 sec. for recognizing protein names in the plain text of 4,259 sentences.

3.2 Dictionary enrichment

The advantage of the dictionary-based statistical approach is that it is versatile, as the user can easily improve its performance with no retraining. We assume the following situation as the ideal case: suppose that a user needs to analyze a large amount of text with protein names. The user wants to know

the maximum performance achievable for identifying protein names with our dictionary-based statistical recognizer which can be achieved by adding more protein names to the current dictionary. Note that protein names should be identified in context. That is, recall of the NER results with the ideal dictionary is not 100%. Some protein names in the ideal dictionary are dropped during statistical tagging or labelling.

Table 2 shows the scores after each step of dictionary enrichment. The first block (Tagging) shows the tagging performance after adding protein names appearing in the *test set* to the dictionary. The second block (Labelling) shows the performance of the sequence labelling of the output of the first step. Note that tagging and the sequence labelling models are not retrained using the test set.

3.3 Discussion

It is not possible in reality to train the recognizer on target data, *i.e.*, the test set, but it would be possible for users to add discovered protein names to the dictionary so that they could improve the overall performance of the recognizer without retraining.

Rule-based and procedural approaches are taken in (Fukuda et al., 1998; Franzen et al., 2002). Machine learning-based approaches are taken in (Collier et al., 2000; Lee et al., 2003; Kazama et al., 2002; Tanabe and Wilbur, 2002; Yamamoto et al., 2003; Tsuruoka, 2006; Okanohara et al., 2006). Machine learning algorithms used in these studies are Naive Bayes, C4.5, Maximum Entropy Models, Support Vector Machines, and Conditional Random Fields. Most of these studies applied machine learning techniques to *tokenized* sentences.

Table 3 shows the scores reported by other systems. Tsai et al. (Tsai et al., 2006) and Zhou and Su (Zhou and Su, 2004) combined machine learning techniques and hand-crafted rules. Tsai et al. (Tsai et al., 2006) applied CRFs to the JNLPBA-2004 data. After applying pattern-based post-processing, they achieved the best F-score (75.12) among those reported so far. Kim and Yoon(Kim and Yoon, 2007) also applied heuristic post-processing. Zhou and Su (Zhou and Su, 2004) achieved an F-score of 73.77.

Purely machine learning-based approaches have been investigated by several researchers. The GENIA Tagger (Tsuruoka, 2006) is trained on the JNLPBA-2004 Corpus. Okanohara et al. (Okanohara et al., 2006) employed semi-Markov CRFs whose performance was evaluated against the JNLPBA-2004 data set. Yamamoto et al. (Yamamoto et al., 2003) used SVMs for character-based protein name recognition and sequential labelling. Their protein name extraction performance was 69%. This paper extends the machine learning approach with a curated dictionary and CRFs and achieved high F-score 73.78, which is the top score among the heuristics-free NER systems. Table 4 shows typical recognition errors found in the recognition results that achieved F-score 73.78. In some cases, protein name boundaries of the JNLPBA-2004 data set are not consistent. It is also one of the reasons for the recognition errors that the data set contains general protein names, such as domain, family, and binding site names as well as anaphoric expressions, which are usually not covered by protein name repositories. Therefore, our impression on the performance is that an F-score of 73.78 is sufficiently high.

Furthermore, thanks to the dictionary-based approach, it has been shown that the upper bound performance using ideal dictionary enrichment, without any retraining of the models, has an F-score of 78.72.

4 Conclusions

This paper has demonstrated how to utilize known named entities to achieve better performance in statistical named entity recognition. We took a two-step approach where sentences are first tokenized and tagged based on a biomedical dictionary that consists of general English words and about 1.3 million protein names. Then, a statistical sequence labelling step predicted protein names that are not listed in the dictionary and, at the same time, reduced false negatives in the POS/PROTEIN tagging results. The significant benefit of this approach is that a user, not a system developer, can easily enhance the performance by augmenting the dictionary. This paper demonstrated that the state-of-the-art F-score 73.78 on the standard JNLPBA-2004 data set was achieved by our approach. Furthermore, thanks to the dictionary-based NER approach, the upper bound performance using ideal dictionary enrichment, without any retraining of the models, yielded F-score 78.72.

5 Acknowledgments

This research is partly supported by EC IST project FP6-028099 (BOOTStrep), whose Manchester team is hosted by the JISC/BBSRC/EPSRC sponsored National Centre for Text Mining.

Table 3: Conventional results for protein name recognition

Authors	R	P	F
Tsai et al.(Tsai et al., 2006)	71.31	79.36	75.12
Our system	**79.85**	**68.58**	**73.78**
Zhou and Su(Zhou and Su, 2004)	69.01	79.24	73.77
Kim and Yoon(Kim and Yoon, 2007)	75.82	71.02	73.34
Okanohara et al.(Okanohara et al., 2006)	77.74	68.92	73.07
Tsuruoka(Tsuruoka, 2006)	81.41	65.82	72.79
Finkel et al.(Finkel et al., 2004)	77.40	68.48	72.67
Settles(Settles, 2004)	76.1	68.2	72.0
Song et al.(Song et al., 2004)	65.50	73.04	69.07
Rössler(Rössler, 2004)	72.9	62.0	67.0
Park et al.(Park et al., 2004)	69.71	59.37	64.12

References

J. Aoe, An Efficient Digital Search Algorithm by Using a Double-Array Structure, *IEEE Transactions on Software Engineering*, 15(9):1066–1077, 1989.

L.E. Baum and T. Petrie, Statistical inference for probabilistic functions of finite state Markov chains, *The Annals of Mathematical Statistics*, 37:1554–1563, 1966.

J. Chang, H. Schutze, R. Altman, GAPSCORE: Finding Gene and Protein names one Word at a Time, *Bioinformatics*, Vol. 20, pp. 216-225, 2004.

N. Collier, C. Nobata, J. Tsujii, Extracting the Names of Genes and Gene Products with a Hidden Markov Model, *Proc. of the 18th International Conference on Computational Linguistics (COLING'2000)*, Saarbrucken, 2000.

Jenny Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Gail Sinclair and Christopher Manning, Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 88–91, 2004.

K. Franzen, G. Eriksson, F. Olsson, L. Asker, P. Liden, and J. Koster, Protein Names and How to Find Them, *Int. J. Med. Inf.*, Vol. 67, pp. 49–61, 2002.

K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi, Toward information extraction: identifying protein names from biological papers, *PSB*, pp. 705-716, 1998.

J. Kazama, T. Makino, Y. Ohta, J. Tsujii, Tuning Support Vector Machines for Biomedical Named Entity Recognition, *Proc. of ACL-2002 Workshop on Natural Language Processing in the Biomedical Domain*, pp. 1–8, 2002.

J.-D. Kim, T. Ohta, Y. Tateisi, J. Tsujii: GENIA corpus - semantically annotated corpus for bio-textmining, *Bioinformatics* 2003, 19:i180-i182.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, Introduction to the Bio-Entity Recognition Task at JNLPBA, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 70–75, 2004.

S. Kim, J. Yoon: Experimental Study on a Two Phase Method for Biomedical Named Entity Recognition, *IEICE Transactions on Informaion and Systems* 2007, E90-D(7):1103–1120.

Taku Kudo and Kaoru Yamamoto and Yuuji Matsumoto, Applying Conditional Random Fields to Japanese Morphological Analysis, *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*, pp. 230–237, 2004.

J. Lafferty, A. McCallum, and F. Pereira, Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, *Proc. of ICML-2001*, pp.282–289, 2001

K. J. Lee, Y. S. Hwang and H. C. Rim (2003), Two-Phase Biomedical NE Recognition based on SVMs, *Proc. of ACL 2003 Workshop on Natural Language Processing in Biomedicine*, Sapporo, 2003.

McCallum A, Freitag D, Pereira F.: Maximum entropy Markov models for information extraction and segmentation, *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000:591-598.

Daisuke, Okanohara, Yusuke Miyao, Yoshimasa Tsuruoka and Jun'ichi Tsujii, Improving the Scalability of Semi-Markov Conditional Random Fields for Named Entity Recognition, *Proc. of ACL 2006*, Sydney, 2006.

Kyung-Mi Park, Seon-Ho Kim, Do-Gil Lee and Hae-Chang Rim. Boosting Lexical Knowledge for Biomedical Named Entity Recognition, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 76-79, 2004.

Marc Rössler, Adapting an NER-System for German to the Biomedical Domain, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 92–95, 2004.

Burr Settles, Biomedical Named Entity Recognition Using Conditional Random Fields and Novel Feature

Table 4: Error Analysis

	False positives		
	Cause	Correct extraction	Identified term
1	dictionary	-	protein, binding sites
2	prefix word	trans-acting factor	common trans-acting factor
3	unknown word	-	ATTTGCAT
4	sequential labelling error	-	additional proteins
5	test set error	-	Estradiol receptors
	False negatives		
	Cause	Correct extraction	Identified term
1	anaphoric	(*the*) receptor, (*the*) binding sites	-
2	coordination (and, or)	transcription factors NF-kappa B and AP-1	transcription factors NF-kappa B
3	prefix word	activation protein-1	protein-1
		catfish STAT	STAT
4	postfix word	nuclear factor kappa B complex	nuclear factor kappa B
5	plural	protein tyrosine kinase(s)	protein tyrosine kinase
6	family name, biding site,	T3 binding sites	-
	and domain	residues 639-656	-
7	sequential labelling error	PCNA	-
		Chloramphenicol acetyltransferase	-
8	test set error	superfamily member	-

Sets, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 104–1007, 2004.

Yu Song, Eunju Kim, Gary Geunbae Lee and Byoungkee Yi, POSBIOTM-NER in the shared task of BioNLP/NLPBA 2004, *Proc. of the Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-2004)*, pp. 100-103, 2004.

L. Tanabe and W. J. Wilbur, Tagging Gene and Protein Names in Biomedical Text, *Bioinformatics*, 18(8), pp. 1124–1132, 2002.

E.F. Tjong Kim Sang and J. Veenstra, Representing Text Chunks,*EACL-99*, pp. 173-179, 1999.

Richard Tzong-Han Tsai, W.-C. Chou, S.-H. Wu, T.-Y. Sung, J. Hsiang, and W.-L. Hsu, Integrating Linguistic Knowledge into a Conditional Random Field Framework to Identify Biomedical Named Entities, *Expert Systems with Applications*, 30 (1), 2006.

Yoshimasa Tsuruoka, GENIA Tagger 3.0, `http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/`, 2006.

K. Yamamoto, T. Kudo, A. Konagaya and Y. Matsumoto, Protein Name Tagging for Biomedical Annotation in Text, in Proc. of ACL-2003 Workshop on Natural Language Processing in Biomedicine, Sapporo, 2003.

Guofeng Zhou and Jian Su, Exploring Deep Knowledge Resources in Biomedical Name Recognition, *Proceedings of the Joint Workshop on Natural Language Processing of Biomedicine and its Applications (JNLPBA-2004)*, pp. 96-99, 2004.

Species Disambiguation for Biomedical Term Identification

Xinglong Wang and Michael Matthews
School of Informatics, University of Edinburgh
2 Buccleuch Place, Edinburgh, EH8 9LW, UK
{xwang,mmatsews}@inf.ed.ac.uk

Abstract

An important task in information extraction (IE) from biomedical articles is term identification (TI), which concerns linking entity mentions (e.g., terms denoting proteins) in text to unambiguous identifiers in standard databases (e.g., RefSeq). Previous work on TI has focused on species-specific documents. However, biomedical documents, especially full-length articles, often talk about entities across a number of species, in which case resolving species ambiguity becomes an indispensable part of TI. This paper describes our rule-based and machine-learning based approaches to species disambiguation and demonstrates that performance of TI can be improved by over 20% if the correct species are known. We also show that using the species predicted by the automatic species taggers can improve TI by a large margin.

1 Introduction

The exponential growth of the amount of scientific literature in the fields of biomedicine and genomics has made it increasingly difficult for scientists to keep up with the state of the art. The TXM project (Alex et al., 2008a), a three-year project which aims to produce software tools to aid curation of biomedical papers, targets this problem and exploits natural language processing (NLP) technology in an attempt to automatically extract enriched protein-protein interactions (EPPI) and tissue expressions (TE) from biomedical text.

A critical task in TXM is term identification (TI), the task of grounding mentions of biomedical named entities to identifiers in referent databases. TI can be seen as an intermediate task that builds on the previous component in an information extraction (IE) pipeline, i.e., named entity recognition (NER), and provides crucial information as input to the more complex module of relation extraction (RE). The structure of the IE pipeline resembles a typical curation process by human biologists. For example, when curating protein-protein interactions (PPIs), a curator would first mark up the protein mentions in text, and then identify the mentions by finding their unique identifiers from standard protein databases such as RefSeq,[1] and finally curate pairs of IDs as PPIs.

TI is a matching and disambiguation process (Wang and Matthews, 2008), and a primary source of ambiguity lies in the model organisms of the terms. In curation tasks, one often needs to deal with collections of articles that involve entities of a large variety of species. For example, our collection of articles from PubMed and PubMed Central involve over 100 model organisms. Also, it is often the case that more than one species appear in the same document, especially when the document is a full-length article. In our dataset, 74% of the articles concern more than one organism. In many standard databases, such as RefSeq and SwissProt, homolog proteins in different species, which often contain nearly identical synonym lists, are assigned distinct identifiers. This makes biomedical terms even more polysemous and hence species disambiguation becomes crucial to TI. For example, querying RefSeq[2] with the protein mention *plk1* resulted in 98

[1] http://www.ncbi.nlm.nih.gov/RefSeq/

[2] The searches were carried out on November 5, 2007.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 71–79,
Columbus, Ohio, USA, June 2008.

hits. By adding a species to the query, e.g. *mouse*, one can significantly reduce the number of results to two.

This paper describes our work on the task of species disambiguation. We also report the performance gain of a TI system from integration of various automatic species taggers. The paper is organised as follows. Section 2 gives a brief overview of related work. Section 3 presents our methodologies for species disambiguation. Section 4 describes a rule-based TI system that we developed in the TXM project, and the evaluation metrics. This section also reports the evaluation results of the TI system with and without help from the species predicted by the taggers. We finally conclude in Section 5.

2 Related Work

The most relevant work to ours are the *Gene Normalisation* (GN) tasks (Morgan and Hirschman, 2007; Hirschman et al., 2004) in the BioCreAtIvE I & II workshops (Hirschman et al., 2007; Hirschman et al., 2005), which provided forums for exchanging thoughts and methodologies on tackling the task of TI. The data provided in the GN tasks, however, were species-specific, which means that the lexicons and datasets were concerned with single model organisms and thus species disambiguation was not required. A few participating systems, however, integrated a filter to rule out entities with erroneous species (Hanisch et al., 2005; Fluck et al., 2007), which were reported to be helpful. Another difference between our task and the BioCreAtIvE GN ones is that we carry out TI on entity level while GN on document level.

It is worth mentioning that the protein-protein interaction task (IPS) in BioCreAtIvE II has taken into account species ambiguity. The IPS task resembles the work-flow of manual curation of PPIs in articles involving multiple species, and to accomplish the task, one would require a full pipeline of IE systems, including named entity recognition, term identification and relation extraction. The best result for IPS (Krallinger et al., 2007) was fairly low at 28.85% $F1$, which reflects the difficulty of the task. Some participants of IPS have reported (e.g., Grover et al., 2007) that resolving species ambiguity was one of the biggest challenges. Our analysis of the IPS training data revealed that the interacting proteins in this corpus belong to over 60 species, and only 56.27% of them are *human*.

As noted in previous work (Krauthammer and Nenadic, 2004; Chen et al., 2005; Krallinger et al., 2007; Wang, 2007), determining the correct species for the protein mentions is a very important step towards TI. However, as far as we know, there has been little work in species disambiguation and in to what extent resolving species ambiguity can help TI.

3 Species Disambiguation

3.1 Data and Ontology

The species tagger was developed on the ITI TXM corpora (Alex et al., 2008b), which were produced as part of the TXM project (Alex et al., 2008a). We created two corpora in slightly different domains, EPPI and TE. The EPPI corpus consists of 217 full-text papers selected from PubMed and PubMed Central and domain experts annotated all documents for both protein entities and PPIs, as well as extra (enriched) information associated with the PPIs and normalisations of the proteins to publicly available ontologies. The TE corpus consists of 230 full-text papers, in which entities such as proteins, tissues, genes and mRNAcDNAs were identified, and a new tissue expression relation was marked up.

We used these corpora to develop a species tagging system. As the biomedical entities in the data were manually assigned with standard database identifiers,[3] it was straightforward to obtain their species IDs through the mappings provided by EntrezGene and RefSeq. In more detail, proteins, protein complexes, genes and mRNAcDNAs in both EPPI and TE datasets were assigned with NCBI Taxonomy IDs (TaxIDs)[4] denoting their species. The EPPI and TE datasets have different distributions of species. The entities in the EPPI data belong to 118 species with *human* being the most frequent at 51.98%. In the TE data, the entities are across 67 species and *mouse* was the most frequent at 44.67%.[5]

To calculate the inter-annotator-agreement, about 40% of the documents were doubly annotated by different annotators. The averaged $F1$ scores of

[3] In our data, *genes* are tagged with EntrezGene IDs, and *proteins* and *mRNAcDNAs* with RefSeq IDs.

[4] `http://www.ncbi.nlm.nih.gov/sites/entrez?db=Taxonomy`

[5] These figures were obtained from the training split of the datasets.

	EPPI devtest			TE devtest		
	P	R	F1	P	R	F1
PreWd	81.88	1.87	3.65	91.49	1.63	3.21
PreWd + Spread	63.85	14.17	23.19	77.84	17.97	29.20
PreWd Sent	60.79	5.16	9.52	56.16	7.76	13.64
PreWd Sent + Spread	39.74	50.54	44.49	31.71	46.68	37.76
Prefix	98.98	3.07	5.96	77.93	2.97	5.72
PreWd + Prefix	91.95	4.95	9.40	82.27	4.62	8.75
PreWd + Prefix + Spread	68.46	17.49	27.87	77.77	21.26	33.39

Table 1: Results (%) of the rule-based species tagger.

species annotation on the doubly annotated EPPI and TE datasets are 86.45% and 95.11%, respectively, indicating that human annotators have high agreement when assigning species to biomedical entities.

3.2 Detecting Species Words

Words referring to species, such as *human*, are important indicators of the species of the nearby entities. We have developed a rule-based program that detects *species words*, which were used to help the species identification systems described in the following sections.

The species word tagger is a lexical look-up component which applies to tokenised text and marks content words such as *human*, *murine* and *D. melanogaster* with their corresponding species TaxIDs. In addition, rules written in an *lxtransduce* grammar[6] are used to identify species prefixes (e.g., 'h' for *human*, 'm' for *mouse*). For example, the term *mSos-1* would be assigned with a TaxID for *mouse*. Note that a species "word" may contain several words, for example, "E. coli". Please see (Wang and Grover, 2008) for more details on the species word tagger.

3.3 Assigning Species to Entities

3.3.1 Rule-based Approach

It is intuitive that a species word that occurs near an entity (e.g., *"mouse p53"*) is a strong indicator of its species. To assess this intuition, we developed a set of five rules using heuristics and species words detected by the species word tagger.

- *PreWd*: If the word preceding an entity is a species word, assign the species indicated by that word to the entity.
- *PreWd Sent*: If a species word that occurs to the left of an entity and in the same sentence, assign the species indicated by that word to the entity.
- *Prefix*: If an entity has a species-indicating prefix, e.g., *mSos-1*, then tag the species to that entity.
- *Spread*: Spread the species of an entity *e* to all entities in the same document that have the same surface form with *e*. This rule must be used in conjunction with the other rules.
- *Majority Vote*:[7] Count the species words in a document and assign as a weight to each species the proportion of all species words in the document that refer to the species.[8] Tag all entities in the document the species with the highest weight, defaulting to *human* in the case of a tie.

Table 1 shows the results of species tagging when the above rules were applied. As we can see, the precision of the systems that rely solely on the previous species words or prefixes is very good but the recall is low. The system that looks at the previous species word in the same sentence does better as measured by $F1$. In addition, spreading the species improves both systems but the overall results are still not satisfactory.

It is slightly counter-intuitive that using a rule such as '*PreWd*' did not achieve perfect precision. Closer inspection revealed that most of the false positives were due to a few problematic guidelines in the annotation process. For example,

- "*The amounts of human and mouse CD200R ...*", where 'CD200R' was tagged as *mouse (10090)* by the system but the gold-standard answer was *human (9606)*. This was due to the fact that the annotation tool was not able to assign multiple correct species

[6] See `http://www.ltg.ed.ac.uk/software/ltxml2` for details of the LT-XML 2 tools developed at the LTG group at Edinburgh University.

[7] The *Majority Vote* rule was used by default in the TI system, which is described in Section 4.1.

[8] For example, if there are N species words in a document and N_{human} are associated with $human$, the $human$ species weight is calculated as $\frac{N_{human}}{N}$.

	BL	EPPI Model	TE Model	Combined Model	EPPI Model +Rules	TE Model +Rules	Combined Model +Rules
EPPI devtest	60.56	73.03	58.67	72.28	**74.24**	59.67	73.77
TE devtest	30.22	67.15	69.82	67.20	67.53	**70.14**	67.47
Overall	48.88	70.77	62.96	70.33	**71.66**	63.70	71.34

Table 2: Accuracy (%) of the machine-learning based species tagger and the hybrid species tagger as tested on the EPPI and TE devtest datasets. An 'Overall' score is the micro-average of a system's accuracy on both datasets.

to a single entity.

- "... *wheat* e*IFiso4G* ...", where 'eIFiso4G' was tagged as *wheat (4565)* but the annotator thought it was *Triticum (4564)*. In this case, TaxID 4565 is a species under genus 4564, and arguably is also a correct answer. Other similar cases include Xenopus vs. Xenopus tropicalis, and Rattus vs. Rattus norvegicus, etc. This is the main cause for the false positives as our system always predicts species instead of genus or TaxIDs of any other ranks, which the annotators occasionally employed.

3.3.2 Machine Learning Approach

We split the EPPI and TE datasets into training and development test (devtest) sets and developed a machine-learning (ML) based species tagger. Using the training splits, we trained a maximum entropy classifier[9] using the following set of features, with respect to each entity occurrence. The parameter n was empirically developed using the training datasets.

- *leftContext* The n word lemmas to the left of the entity, without position ($n = 200$).
- *rightContext* The n word lemmas to the right of the entity, without position ($n = 200$).
- *leftSpeciesIDs* The n species IDs, located to the left of the entity and assigned by the species word tagger ($n = 5$).
- *rightSpeciesIDs* The n species IDs, located to the right of the entity and assigned by the species word tagger ($n = 5$).
- *leftNouns* The n nouns to the left of the entity (with order and $n = 2$). This feature attempts to capture cases where a noun preceding an entity indicates species, e.g., *mouse protein p53*.
- *leftAdjs* The n adjectives to the left of the entity (with order and $n = 2$). This feature intends to capture cases where an adjective preceding an entity indicates species, e.g., *murine protein p53*.
- *leftSpeciesWords* The n species word forms, identified by the species word tagger, located to the left of the entity ($n = 5$).
- *rightSpeciesWords* The n species word forms, identified by the species word tagger, located to the right of the entity ($n = 5$).
- *firstLetter* The first character of the entity itself. Sometimes the first letters of entities indicate their species, e.g., *hP53*.
- *documentSpeciesIDs* All species IDs that occur in the article in question.
- *useStopWords* If this feature is switched on then filter out the words that appear in a pre-compiled stopword list from the above features. The list consists of frequent common English words such as prepositions (e.g., *in*).
- *useStopPattern* If this feature is switched on then filter out the words consisting only of digits and punctuation characters.

The results of the ML species tagger are shown in Table 2. We measure the performance in accuracy instead of $F1$ because the ML based tagger assigns a species tag to *every* entity occurrence, and therefore precision is equal to recall. We tested four models on the devtest portions of the EPPI and TE corpora:

- *BL*: a baseline system, which tags the devtest instances using the most frequent species occurring in the corresponding training dataset. For example, *human* is the most frequent species in the EPPI training data, and therefore all entities in the EPPI devtest dataset were tagged with *human*.
- EPPI *Model*: obtained by training the maxent classifier on the EPPI training data.
- TE *Model*: obtained by training the maxent classifier on the TE training data.
- *Combined Model*: obtained by training the maxent classifier on a joint dataset consisting of both the EPPI and TE training corpora.

3.3.3 Hybrid Approach

As we have shown, rules '*PreWd*' and '*Prefix*' achieved very good precision but low recall, which

[9] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

suggests that when these rules were applicable, it is highly likely that they would get the correct species. Based on this observation, we combined the ML approach and the rule-based approach in such a way that the rules '*PreWd*' and '*Prefix*' were applied on top of ML and override predictions made by ML. In other words, the rules act as a post-processor and correct the decisions made by the ML when very strong species indicators such as previous species words or species prefixes are detected. This should increase precision and at the same time keep recall relatively intact. The hybrid systems were tested on the same datasets and the results are shown in the right 3 columns in Table 2.

We performed significance tests on the results in Table 2. First, a Friedman test was used to determine whether the 7 sets of results[10] were significantly different, and then pairwise Wilcoxon Signed Rank tests were employed to tell whether any system performed significantly better than others. On both datasets, the 6 machine-learning models significantly outperformed the baseline ($p < 0.01$). On EPPI devtest dataset, the EPPI models (with or without rules) and the Combined Models outperformed the TE models ($p < 0.05$), while on TE dataset, the TE models and the Combined Models outperformed the EPPI models ($p < 0.05$). Also, applying the post filtering rules did not significantly improve the ML models, although it appears that adding the rules consistently increase the accuracy by a small margin.

4 Term Identification

4.1 The TI system

The TI system is composed of a matcher which determines a list of candidate identifiers and a ranker that assigns a confidence value to each identifier that is used to rank the candidates in order with the most likely identifiers occurring first. The matcher is based largely on the rule-based system described in (Wang and Matthews, 2008), but has been put into a more flexible framework that allows for defining and customising the rules in a configuration file. In addition, the system has been expanded to perform TI on additional entity types. The rules for each entity were developed using the training data and a visualisation system that compared the synonym list for the target identifiers with the actual entity mentions and provided visual feedback on the true positives and false positives resulting from candidate rules sets. Examples of some of the rules that can be incorporated into the system are listed below. A confidence value is assigned to each of the rules using heuristics and passed to the ranking system.

1. *LowerCase*: Convert the entity mention to lowercase and look up the result in a lower case version of the entity term database.
2. *Norm*: Normalise the mention[11] and look up the result in a normalised version of the term database.
3. *Prefix*: Add and/or remove a set of prefixes from the entity mention and look up the result in the entity term database. The actual prefixes and whether to add or remove them are specified in the configuration file.
4. *Suffix*: Add and/or remove a set of suffixes from the entity mention and look up the result in the entity term database. The actual suffixes and whether to add or remove them are specified in the configuration file.
5. *Porter*: Compute the Porter stem of the entity mention and looked up the synonym in a Porter stemmed version of the entity term database.

The ranking system currently works by defining a set of confidence indicators for each entity, computing the confidence for each indicator and then multiplying each individual confidence together to determine the overall identifier confidence. The following indicators are currently used by the system.

1. *Match*: The confidence as determined by the matcher.
2. *Species*: The confidence that the species of the identifier is the correct species.
3. *Reference Count*: Based on the number of literature references[12] associated with each identifier. The higher the reference count, the higher the confidence.

[10] The Friedman test requires accuracy figures with respect to each document in the datasets, which are not shown in Table 2.

[11] Normalising a string involves converting Greek characters to English (e.g., $\alpha \rightarrow$alpha), converting to lowercase, changing sequential indicators to integer numerals (e.g., *i*, *a*, *alpha*$\rightarrow$1, etc.) and removing all spaces and punctuation. For example, *rab1*, *rab-1*, *rab*α, *rab I* are all normalised to *rab1*.

[12] The Reference Counts were obtained from EntrezGene and RefSeq databases.

4. *Primary Name*: Based on a determination that the entity mention is the primary name for the identifier. This is based both on a name provided by the lexicon and a name derived from the synonym list.

Among these, one of the most critical indicators is the species confidence. By default, this confidence is set to the weight assigned to the species by the *Majority Vote* tagger (see Section 3.3.1). When the species of an entity is tagged by an external species tagger or by human annotators, the default confidence can be overridden. This setting allows us to integrate automatic species taggers, such as the ones described in the previous section, for achieving better TI performance. For example, suppose we want to employ the *Hybrid* species tagger. To compute the species confidence, first the hybrid tagger is used to predict the most likely species and the *Majority Vote* tagger is run at the same time. If the species of an identifier matches the species assigned by the hybrid tagger, the species confidence is set to the weight generated by the hybrid tagger. Otherwise, the confidence is set to the weight generated by the *Majority Vote* tagger.

To assess how much species ambiguity accounts for the overall ambiguity in biomedical entities, we estimated the averaged *ambiguity rates* for the protein entities in the TXM datasets, without and with the species information. Suppose there are n unique protein mentions in a dataset. First, we look up the RefSeq database by exact match with every unique protein mention m_i, where $i \in \{0..n-1\}$, and for each m_i we retrieve two lists of identifiers: L_i and L'_i, where L_i consists of all identifiers and L'_i only contains the identifiers whose model organism matches the manually tagged species of the protein mention. The ambiguity rates without and with species are computed by $\frac{\sum_{i=0}^{n-1}|L_i|}{n}$ and $\frac{\sum_{i=0}^{n-1}|L'_i|}{n}$, respectively. Table 3 shows the ambiguity rates on the EPPI and TE datasets.

	Protein Cnt	ID Cnt	Ambiguity
EPPI	6,955	184,633	26.55
EPPI species	6,955	17,357	2.50
TE	8,539	103,016	12.06
TE species	8539	12,705	1.49

Table 3: Ambiguity in protein entities, with and without species information, in EPPI and TE datasets.

4.2 Experiments on TXM Data

To identify whether species disambiguation can improve performance of TI, we ran the TI system on the EPPI and TE data. As shown in Tables 4 and 5, we tested the TI systems with or without help from a number of species tagging systems, including:

- *Baseline*: Run TI without species tags.[13]
- *Gold Species*: Run TI with manually tagged species. This is the upper-bound performance.
- *Rule*: Run TI with species predicted by the rule-based species tagger.
- *ML(human/mouse)*: Run TI with the species that occurs most frequently in the training datasets (i.e., *human* for EPPI and *mouse* for TE).
- *ML(*EPPI*)*: Run TI with species predicted by the ML tagger trained on the EPPI training dataset.
- *ML(*EPPI*)+Rule*: Run TI with species predicted by the hybrid system using both ML(EPPI) and the rules.
- *ML(*TE*)*: Run TI with species predicted by the ML tagger trained on the TE training dataset.
- *ML(*TE*)+Rule*: Run TI with species predicted by the hybrid system using both ML(TE) and the rules.
- *ML(*EPPI+TE*)*: Run TI with species predicted by the ML tagger trained on both EPPI and TE training data.
- *ML(*EPPI+TE*)+Rule*: Run TI with species predicted by the hybrid system using both ML(EPPI+TE) and the rules.

We score the systems using *t*op n precision, where $n \in \{1, 5, 10, 15, 20\}$. The argument for this evaluation scheme is that if a TI system is not good enough in predicting a single identifier correctly, a 'bag' of IDs with the correct answer included would also be helpful. The 'Avg. Rank' field denotes the averaged position where the correct answer lies in, and the lower the value is, the better the TI system performs. For example, a TI system with an 'Avg. Rank' of 1 would be ideal, as it would always return the correct ID at the top of the list. Note that in the TE data, not only protein entities, but also genes, MRNACDNA, and GOMOPS[14] were tagged.

On both datasets, using the gold standard species much improved accuracy of TI (e.g., 19.2% on EPPI

[13] Note that the TI system already integrated a basic species tagging system that uses the *Majority Vote* rule as described in Section 3.3.1. Thus this is a fairly high 'baseline'.

[14] GOMOP is a tag that denotes an entity being either a gene, or an MRNACDNA, or a protein, which was used when the annotator could not determine what type the entity in question was.

Method	Prec@1	Prec@5	Prec@10	Prec@15	Prec@20	Avg. Rank
Baseline	54.31	73.45	76.44	77.90	78.51	5.82
Gold Species	73.52	79.36	80.75	80.75	80.99	1.62
Rule	54.99	73.72	76.45	77.91	78.52	5.79
ML(human)	65.66	76.36	78.82	79.78	80.03	2.58
ML(EPPI)	65.24	76.82	79.01	79.93	80.29	2.39
ML(EPPI)+Rule	**65.88**	**77.09**	**79.04**	**79.94**	80.30	**2.36**
ML(TE)	55.87	75.14	78.69	79.85	80.30	2.86
ML(TE)+Rule	56.54	75.47	78.70	79.86	80.31	2.83
ML(EPPI+TE)	64.55	76.48	78.53	79.83	80.38	2.49
ML(EPPI+TE)+Rule	65.03	76.62	78.55	79.84	**80.39**	2.46

Table 4: Results of TI on the EPPI dataset. All figures, except 'Avg. Rank', are percentages. This evaluation was carried out on protein entities only.

Method	Prec@1	Prec@5	Prec@10	Prec@15	Prec@20	Avg. Rank
Baseline	63.24	76.20	77.30	77.94	78.25	1.72
Gold Species	71.82	78.03	78.34	78.40	78.41	1.29
Rule	63.45	76.21	77.30	**77.95**	**78.25**	1.72
ML(mouse)	58.76	75.40	77.25	77.92	78.24	1.90
ML(EPPI)	66.59	76.53	77.23	77.76	78.12	1.68
ML(EPPI)+Rule	**66.85**	**76.54**	77.24	77.76	78.12	**1.67**
ML(TE)	66.12	76.25	77.32	77.81	78.11	1.70
ML(TE)+Rule	66.37	76.25	**77.32**	77.81	78.11	1.70
ML(EPPI+TE)	65.78	76.14	77.28	77.84	78.12	1.71
ML(EPPI+TE)+Rule	66.03	76.14	77.29	77.84	78.12	1.70

Table 5: Results of TI on the TE dataset. All figures, except 'Avg. Rank', are percentages. There are four entity types in the TE data, i.e., protein, gene, mRNAcDNA and GOMOP. The evaluation was carried out on all entity types.

data). Also, automatically predicted species tags were proven to be helpful. On the EPPI data, the *ML(*EPPI*)+Rule* outperformed other systems. Note that the species distribution in the devtest dataset is strongly biased to *human*, which explains why the *ML(human)* system performed nearly as well. However, defaulting to *human* was not guaranteed to succeed because one would not be able to know the prior species in a collection of unseen documents. Indeed, on the TE data, the system *ML(mouse)*, which uses the most frequent species in the training data, i.e. *mouse*, as default, yielded poor results.

4.3 Experiments on BioCreAtIvE Data

To assess the portability of the species tagging approaches, an "artificial" dataset was created by joining the species-specific datasets from BioCreAtIvE 1 & 2 GN tasks to form a corpus consisting of four species. In detail, four datasets were taken, three from BioCreAtIvE 1 task 1B (i.e., fly, mouse and yeast) and one from BioCreAtIvE 2 task GN (i.e., human). Assuming genes in each dataset are species-specific,[15] we can train/test ML models for species disambiguation and apply them to help TI. This task is more difficult than the original BioCreAtIvE GN tasks due to the additional ambiguity caused by multiple model organisms.

We first carried out experiments on species disambiguation. In addition to the TXM (i.e., the system uses ML(EPPI+TE)+Rule model) and the *Majority Vote* taggers, we trained the species tagger on a dataset comprising of the devtest sets from the BioCreAtIvE I & II GN tasks. In more detail, we first pre-processed the dataset and marked up gene entities with an NER system (Alex et al., 2007; Grover et al., 2007).[16] The entities were also tagged with the

[15]This assumption is not strictly true because each dataset may contain genes of other species, and it would be hard to assess how true it is as abstracts in the BioCreAtIvE GN datasets are not normalised to an entity level.

[16]The NER system was trained on BioCreAtIvE II GM training and test datasets.

species as indicated by the source dataset where they were drawn from, which were used as the 'Gold' species. Using the same algorithm and feature set as described in Section 3.3.2, a *BC model* was trained.

	human	fly	mouse	yeast
Majority Vote	82.35	78.43	71.69	85.12
BC model	70.23	89.24	75.41	87.64
TXM model	93.35	3.27	31.89	3.49

Table 6: Accuracy (%) of the species disambiguation systems as tested on the BioCreAtIvE I & II test data. The 'BC model' was trained on the BioCreAtIvE devtest data, the 'TXM model' was trained on the TXM EPPI and TE training data, and the '*Majority Vote*' was the default species tagging system in the TI system (see Section 3.3.1).

As shown in Table 6, except on *human*, the TXM model yielded very disappointing results, whereas the BC model did well overall. This was because the TXM model was trained on a dataset where *fly* and *yeast* entities occur rarely with only 2% and 5% of the training instances belonging to these species, respectively, which again revealed the influence of the bias introduced in the training material to the ML models.

System	Precision	Recall	$F1$
Gold	70.1	63.3	66.5
Majority Vote	46.7	56.3	51.0
TXM model	37.8	46.5	41.7
BC model	45.8	56.1	50.4

Table 7: Performance of TI with or without the automatically predicted species on the joint BioCreAtIvE GN test dataset.

Using the species disambiguation models, we carried out TI experiments, using the same procedure as we did on the TXM data. The results were obtained using the official BioCreAtIvE GN scorers[17] and are presented in Table 7. Performance of TI assisted by all three species taggers were much behind that of TI using the gold-standard species, which shows species-tagging can potentially enhance TI performance and there is much room for improving the species disambiguation systems. On the other hand, it was disappointing that the '*Majority Vote*' system, which did not use any external species tagger, achieved the best results, while TI with the 'BC model' tagger yielded slightly worse results and the TXM model performed poorly.

# Species	# of Docs	% of Docs
1	96	26.20
2	121	32.79
3+	153	41.19

Table 8: # of species per document in the TXM data.

One possible reason that the '*Majority Vote*' tagger yielded reasonably good result on the BioCreAtIvE dataset, but unsatisfactory result on the TXM datasets was due to the difference in document length in the two corpora: the BioCreAtIvE corpus is comprised of abstracts and the TXM corpora consist of only full-length articles. In abstracts, authors are inclined to only talk about the main biomedical entities described in the paper, whereas in full articles, they tend to describe a larger number of entities, possibly in multiple species, for the purposes of describing related work or comparison. Recall that the '*Majority Vote*' rule outputs the species indicated by the majority of the species words, which would obviously perform better on abstracts, where more likely only one species is described, than on full-length articles. Table 8 shows the number of species per document in the TXM data, where most documents (i.e., 74%) involve more than one species, in which cases the '*Majority Vote*' would not be able to take obvious advantage.

5 Conclusions and Future Work

This paper presented a range of solutions to the task of species disambiguation and evaluated their performance on the ITI TXM corpus, and on a joint dataset from BioCreAtIvE I & II GN tasks. We showed that rule-based species tagging systems that exploit heuristics, such as previous species words or species prefix, can achieve very high precision but low recall. ML species taggers, on the other hand, can achieve good overall performance, under the condition that the species distributions in training and test datasets are not too distant. Our best performing species tagger is a hybrid system that first

[17] We tested the TI system on the four original BioCreAtIvE GN datasets separately and the averaged performance was about the median among the participating systems in the workshops. We did not optimise the TXM TI system on BioCreAtIvE, as our point here is to measure the TI performance with or without help from the automatic predicted species.

uses ML to predict species and then applies certain rules to correct errors.

We also performed TI experiments with help from species tags assigned by human annotators, or predicted by the automatic species taggers. On all datasets, the gold-standard species tags improved TI performance by a large margin: 19.21% on the EPPI devtest set, 8.59% on the TE devtest set, and 23.4% on the BioCreAtIvE GN test datasets, which clearly shows that species information is indeed very important for TI. On the EPPI and TE datasets, the species predicted by the best-performing hybrid system improved TI by 11.57% and 3.61%, respectively. On the combined dataset from BioCreAtIvE GN tasks, however, it did not work as well as expected.

In the future we plan to work on better ways to integrate the machine learning approaches and the rules. In particular, we would like to explore statistical relational learning, which may provide ways to integrate rules as constraints into machine learning and may be able to alleviate the bias in the learnt models.

Acknowledgements

The work was supported by the ITI Life Sciences Text Mining programme.[18]

References

B. Alex, B. Haddow, and C. Grover. 2007. Recognising nested named entities in biomedical text. In *Proceedings of BioNLP 2007*, Prague, Czech Republic.

B. Alex, C. Grover, B. Haddow, M. Kabadjov, E. Klein, M. Matthews, S. Roebuck, R. Tobin, and X. Wang. 2008a. Assisted curation: does text mining really help? In *Proceedings of PSB*.

B. Alex, C. Grover, B. Haddow, M. Kabadjov, E. Klein, M. Matthews, S. Roebuck, R. Tobin, and X. Wang. 2008b. The ITI TXM corpus: Tissue expression and protein-protein interactions. In *Proceedings of the LREC Workshop on Building and Evaluating Resources for Biomedical Text Mining*, Morocco.

L. Chen, H. Liu, and C. Friedman. 2005. Gene name ambiguity of eukaryotic nomenclatures. *Bioinformatics*, 21(2):248–256.

J. Fluck, H. Mevissen, H. Dach, M. Oster, and M. Hofmann-Apitius. 2007. ProMiner: Recognition of human gene and protein names using regularly updated disctionaries. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*.

C. Grover, B. Haddow, E. Klein, M. Matthews, L. A. Nielsen, R. Tobin, and X. Wang. 2007. Adapting a relation extraction pipeline for the BioCreAtIvE II task. In *Proceedings of the BioCreAtIvE II Workshop 2007*, Madrid.

D. Hanisch, K. Fundel, H-T Mevissen, R Zimmer, and J Fluck. 2005. ProMiner: Organism-specific protein name detection using approximate string matching. *BMC Bioinformatics*, 6(Suppl 1):S14.

L. Hirschman, M. Colosimo, A. Morgan, J. Columbe, and A. Yeh. 2004. Task 1B: Gene list task BioCreAtIve workshop. In *BioCreative: Critical Assessment for Information Extraction in Biology*.

L. Hirschman, A. Yeh, C. Blaschke, and A. Valencia. 2005. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6(Suppl1):S1.

L. Hirschman, M. Krallinger, and A. Valencia, editors. 2007. *Second BioCreative Challenge Evaluation Workshop*. Fundación CNIO Carlos III, Madrid.

M. Krallinger, F. Leitner, and A. Valencia. 2007. Assessment of the second BioCreative PPI task: Automatic extraction of protein-protein interactions. In *Proceedings of the BioCreAtIvE II Workshop 2007*, pages 41–54, Madrid, Spain.

M. Krauthammer and G. Nenadic. 2004. Term identification in the biomedical literature. *Journal of Biomedical Informatics (Special Issue on Named Entity Recogntion in Biomedicine)*, 37(6):512–526.

A. A. Morgan and L. Hirschman. 2007. Overview of BioCreative II gene normalisation. In *Proceedings of the BioCreAtIvE II Workshop*, Madrid.

X. Wang and C. Grover. 2008. Learning the species of biomedical named entities from annotated corpora. In *Proceedings LREC2008*, Marrakech, Morocco.

X. Wang and M. Matthews. 2008. Comparing usability of matching techniques for normalising biomedical named entities. In *Proceedings of PSB*.

X. Wang. 2007. Rule-based protein term identification with help from automatic species tagging. In *Proceedings of CICLING 2007*, pages 288–298, Mexico City.

[18] `http://www.itilifesciences.com`

Knowledge Sources for Word Sense Disambiguation of Biomedical Text

Mark Stevenson, Yikun Guo
and **Robert Gaizauskas**
Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello Street
Sheffield, S1 4DP
United Kingdom
{inital.surname}@dcs.shef.ac.uk

David Martinez
Department of Computer Science
& Software Engineering
University of Melbourne
Victoria 3010
Australia
davidm@csse.unimelb.edu.au

Abstract

Like text in other domains, biomedical documents contain a range of terms with more than one possible meaning. These ambiguities form a significant obstacle to the automatic processing of biomedical texts. Previous approaches to resolving this problem have made use of a variety of knowledge sources including linguistic information (from the context in which the ambiguous term is used) and domain-specific resources (such as UMLS). In this paper we compare a range of knowledge sources which have been previously used and introduce a novel one: MeSH terms. The best performance is obtained using linguistic features in combination with MeSH terms. Results from our system outperform published results for previously reported systems on a standard test set (the NLM-WSD corpus).

1 Introduction

The number of documents discussing biomedical science is growing at an ever increasing rate, making it difficult to keep track of recent developments. Automated methods for cataloging, searching and navigating these documents would be of great benefit to researchers working in this area, as well as having potential benefits to medicine and other branches of science. Lexical ambiguity, the linguistic phenomena where a word or phrase has more than one potential meaning, makes the automatic processing of text difficult. For example, "cold" has six possible meanings in the Unified Medical Language System (UMLS) Metathesaurus (Humphreys et al., 1998) including "common cold", "cold sensation" and "Chronic Obstructive Airway Disease (COLD)". The NLM Indexing Initiative (Aronson et al., 2000) attempted to automatically index biomedical journals with concepts from the UMLS Metathesaurus and concluded that lexical ambiguity was the biggest challenge in the automation of the indexing process. Weeber et al. (2001) analysed MEDLINE abstracts and found that 11.7% of phrases were ambiguous relative to the UMLS Metathesaurus.

Word Sense Disambiguation (WSD) is the process of resolving lexical ambiguities. Previous researchers have used a variety of approaches for WSD of biomedical text. Some of them have taken techniques proven to be effective for WSD of general text and applied them to ambiguities in the biomedical domain, while others have created systems using domain-specific biomedical resources. However, there has been no direct comparison of which knowledge sources are the most useful or whether combining a variety of knowledge sources, a strategy which has been shown to be successful for WSD in the general domain (Stevenson and Wilks, 2001), improves results.

This paper compares the effectiveness of a variety of knowledge sources for WSD in the biomedical domain. These include features which have been commonly used for WSD of general text as well as information derived from domain-specific resources. One of these features is MeSH terms, which we find to be particularly effective when combined with generic features.

The next section provides an overview of various approaches to WSD in the biomedical domain. Sec-

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 80–87,
Columbus, Ohio, USA, June 2008.

tion 3 outlines our approach, paying particular attention to the range of knowledge sources used by our system. An evaluation of this system is presented in Section 4. Section 5 summarises this paper and provides suggestions for future work.

2 Previous Work

WSD has been actively researched since the 1950s and is regarded as an important part of the process of understanding natural language texts.

2.1 The NLM-WSD data set

Research on WSD for general text in the last decade has been driven by the SemEval evaluation frameworks[1] which provide a set of standard evaluation materials for a variety of semantic evaluation tasks. At this point there is no specific collection for the biomedical domain in SemEval, but a test collection for WSD in biomedicine was developed by Weeber et al. (2001), and has been used as a benchmark by many independent groups. The UMLS Metathesaurus was used to provide a set of possible meanings for terms in biomedical text. 50 ambiguous terms which occur frequently in MEDLINE were chosen for inclusion in the test set. 100 instances of each term were selected from citations added to the MEDLINE database in 1998 and manually disambiguated by 11 annotators. Twelve terms were flagged as "problematic" due to substantial disagreement between the annotators. There are an average of 2.64 possible meanings per ambiguous term and the most ambiguous term, "cold" has five possible meanings. In addition to the meanings defined in UMLS, annotators had the option of assigning a special tag ("none") when none of the UMLS meanings seemed appropriate.

Various researchers have chosen to evaluate their systems against subsets of this data set. Liu et al. (2004) excluded the 12 terms identified as problematic by Weeber et al. (2001) in addition to 16 for which the majority (most frequent) sense accounted for more than 90% of the instances, leaving 22 terms against which their system was evaluated. Leroy and Rindflesch (2005) used a set of 15 terms for which the majority sense accounted for less than 65% of the instances. Joshi et al. (2005) evaluated against the set union of those two sets, providing 28 ambiguous terms. McInnes et al. (2007) used the set intersection of the two sets (dubbed the "common subset") which contained 9 terms. The terms which form these various subsets are shown in Figure 1.

The 50 terms which form the NLM-WSD data set represent a range of challenges for WSD systems. The Most Frequent Sense (MFS) heuristic has become a standard baseline in WSD (McCarthy et al., 2004) and is simply the accuracy which would be obtained by assigning the most common meaning of a term to all of its instances in a corpus. Despite its simplicity, the MFS heuristic is a hard baseline to beat, particularly for unsupervised systems, because it uses hand-tagged data to determine which sense is the most frequent. Analysis of the NLM-WSD data set showed that the MFS over all 50 ambiguous terms is 78%. The different subsets have lower MFS, indicating that the terms they contain are more difficult to disambiguate. The 22 terms used by (Liu et al., 2004) have a MFS of 69.9% while the set used by (Leroy and Rindflesch, 2005) has an MFS of 55.3%. The union and intersection of these sets have MFS of 66.9% and 54.9% respectively.

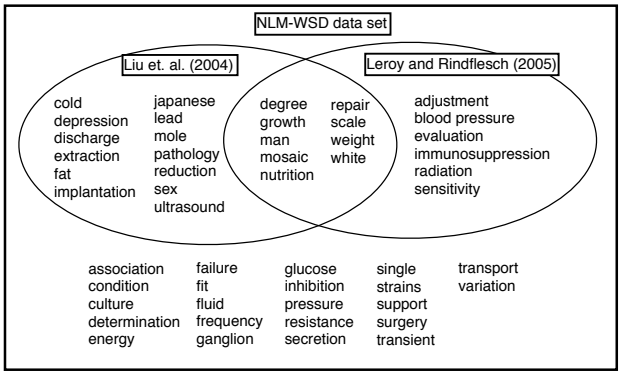

Figure 1: The NLM-WSD test set and some of its subsets. Note that the test set used by (Joshi et al., 2005) comprises the set union of the terms used by (Liu et al., 2004) and (Leroy and Rindflesch, 2005) while the "common subset" is formed from their intersection.

2.2 WSD of Biomedical Text

A standard approach to WSD is to make use of supervised machine learning systems which are trained on examples of ambiguous words in context along with the correct sense for that usage. The

[1] http://www.senseval.org

models created are then applied to new examples of that word to determine the sense being used.

Approaches which are adapted from WSD of general text include Liu et al. (2004). Their technique uses a supervised learning algorithm with a variety of features consisting of a range of collocations of the ambiguous word and all words in the abstract. They compared a variety of supervised machine learning algorithms and found that a decision list worked best. Their best system correctly disambiguated 78% the occurrences of 22 ambiguous terms in the NLM-WSD data set (see Section 2.1).

Joshi et al. (2005) also use collocations as features and experimented with five supervised learning algorithms: Support Vector Machines, Naive Bayes, decision trees, decision lists and boosting. The Support Vector Machine performed scoring 82.5% on a set of 28 words (see Section 2.1) and 84.9% on the 22 terms used by Liu et al. (2004). Performance of the Naive Bayes classifier was comparable to the Support Vector Machine, while the other algorithms were hampered by the large number of features.

Examples of approaches which have made use of knowledge sources specific to the biomedical domain include Leroy and Rindflesch (2005), who relied on information from the UMLS Metathesaurus assigned by MetaMap (Aronson, 2001). Their system used information about whether the ambiguous word is the head word of a phrase identified by MetaMap, the ambiguous word's part of speech, semantic relations between the ambiguous words and surrounding words from UMLS as well as semantic types of the ambiguous word and surrounding word. Naive Bayes was used as a learning algorithm. This approach correctly disambiguated 65.6% of word instances from a set of 15 terms (see Section 2.1). Humphrey et al. (2006) presented an unsupervised system that also used semantic types. They constructed semantic type vectors for each word from a large collection of MEDLINE abstracts. This allowed their method to perform disambiguation at a coarser level, without the need for labeled training examples. In most cases the semantic types can be mapped to the UMLS concepts but not for five of the terms in the NLM-WSD data set. Humphrey et al. (2006) reported 78.6% accuracy over the remaining 45. However, their approach could not be applied to all instances of ambiguous terms and, in particular, is unable to model the "none" tag. Their system could only assign senses to an average of 54% of the instances of each ambiguous term.

McInnes et al. (2007) made use of Concept Unique Identifiers (CUIs) from UMLS which are also assigned by MetaMap. The information contained in CUIs is more specific than in the semantic types applied by Leroy and Rindflesch (2005). For example, there are two CUIs for the term "culture" in UMLS: "C0010453: Anthropological Culture" and "C0430400: Laboratory Culture". The semantic type for the first of these is "Idea or Concept" and "Laboratory Procedure" for the second. McInnes et al. (2007) were interested in exploring whether the more specific information contained in CUIs was more effective than UMLS semantic types. Their best result was reported for a system which represented each sense by all CUIs which occurred at least twice in the abstract surrounding the ambiguous word. They used a Naive Bayes classifier as the learning algorithm. McInnes et al. (2007) reported an accuracy of 74.5% on the set of ambiguous terms tested by Leroy and Rindflesch (2005) and 80.0% on the set used by Joshi et al. (2005). They concluded that CUIs are more useful for WSD than UMLS semantic types but that they are not as robust as features which are known to work in general English, such as unigrams and bigrams.

3 Approach

Our approach is to adapt a state-of-the-art WSD system to the biomedical domain by augmenting it with additional domain-specific and domain-independent knowledge sources. Our basic system (Agirre and Martínez, 2004) participated in the Senseval-3 challenge (Mihalcea et al., 2004) with a performance close to the best system for the English and Basque lexical sample tasks. The system is based on a supervised learning approach. The features used by Agirre and Martínez (2004) are derived from text around the ambiguous word and are domain independent. We refer to these as *linguistic* features. This feature set has been adapted for the disambiguation of biomedical text by adding further linguistic features and two different types of domain-specific features: CUIs (as used by (McInnes et al., 2007)) and Medical Subject Heading (MeSH) terms.

3.1 Features

Our feature set contains a number of parameters which were set empirically (e.g. threshold for unigram frequency in the linguistic features). In addition, we use the entire abstract as the context of the ambiguous term for relevant features rather than just the sentence containing the term. Effects of varying these parameters are consistent with previous results (Liu et al., 2004; Joshi et al., 2005; McInnes et al., 2007) and are not reported in this paper.

Linguistic features: The system uses a wide range of domain-independent features which are commonly used for WSD.

- Local collocations: A total of 41 features which extensively describe the context of the ambiguous word and fall into two main types: (1) bigrams and trigrams containing the ambiguous word constructed from lemmas, word forms or PoS tags[2] and (2) preceding/following lemma/word-form of the content words (adjective, adverb, noun and verb) in the same sentence with the target word. For example, consider the sentence below with the target word *adjustment*.

  > "Body surface area *adjustments* of initial heparin dosing..."

  The features would include the following: left-content-word-lemma "*area adjustment*", right-function-word-lemma "*adjustment of*", left-POS "NN NNS", right-POS "NNS IN", left-content-word-form "*area adjustments*", right-function-word-form "*adjustment of*", etc.

- Syntactic Dependencies: These features model longer-distance dependencies of the ambiguous words than can be represented by the local collocations. Five relations are extracted: object, subject, noun-modifier, preposition and sibling. These are identified using heuristic patterns and regular expressions applied to PoS tag sequences around the ambiguous word. In the above example, "heparin" is noun-modifier feature of "adjustment".

[2] A maximum-entropy-based part of speech tagger was used (Ratnaparkhi, 1996) without the adaptation to the biomedical domain.

- Salient bigrams: Salient bigrams within the abstract with high log-likelihood scores, as described by Pedersen (2001).

- Unigrams: Lemmas of unigrams which appear more frequently than a predefined threshold in the entire corpus, excluding those in a list of stopwords. We empirically set the threshold to 1. This feature was not used by Agirre and Martínez (2004), but Joshi et al. (2005) found them to be useful for this task.

Concept Unique Identifiers (CUIs): We follow the approach presented by McInnes et al. (2007) to generate features based on UMLS Concept Unique Identifiers (CUIs). The MetaMap program (Aronson, 2001) identifies all words and terms in a text which could be mapped onto a UMLS CUI. MetaMap does not disambiguate the senses of the concepts, instead it enumerates all the possible combinations of the concept names found. For example, MetaMap will segment the phrase "Body surface area adjustments of initial heparin dosing ..." into two chunks: "Body surface area adjustments" and "of initial heparin dosing". The first chunk will be mapped onto four CUIs with the concept name "Body Surface Area": "C0005902: Diagnostic Procedure" and "C1261466: Organism Attribute" and a further pair with the name "Adjustments": "C0456081: Health Care Activity" and "C0871291: Individual Adjustment". The final results from MetaMap for the first chunk will be eight combinations of those concept names, e.g. first four by second two concept names. CUIs which occur more than three times in the abstract containing the ambiguous word are included as features.

Medical Subject Headings (MeSH): The final feature is also specific to the biomedical domain. Medical Subject Headings (MeSH) (Nelson et al., 2002) is a controlled vocabulary for indexing biomedical and health-related information and documents. MeSH terms are manually assigned to abstracts by human indexers. The latest version of MeSH contains over 24,000 terms organised into an 11 level hierarchy.

The terms assigned to the abstract in which each ambiguous word occurs are used as features. For example, the abstract containing our example phrase has been assigned 16 MeSH

terms including "M01.060.116.100: Aged", "M01.060.116.100.080: Aged, 80 and over", "D27.505.954.502.119: Anticoagulants" and "G09.188.261.560.150: Blood Coagulation". To our knowledge MeSH terms have not been previously used as a feature for WSD of biomedical documents.

3.2 Learning Algorithms

We compared three machine leaning algorithms which have previously been shown to be effective for WSD tasks.

The **Vector Space Model** is a memory-based learning algorithm which was used by (Agirre and Martínez, 2004). Each occurrence of an ambiguous word is represented as a binary vector in which each position indicates the occurrence/absence of a feature. A single centroid vector is generated for each sense during training. These centroids are compared with the vectors that represent new examples using the cosine metric to compute similarity. The sense assigned to a new example is that of the closest centroid.

The **Naive Bayes** classifier is based on a probabilistic model which assumes conditional independence of features given the target classification. It calculates the posterior probability that an instance belongs to a particular class given the prior probabilities of the class and the conditional probability of each feature given the target class.

Support Vector Machines have been widely used in classification tasks. SVMs map feature vectors onto a high dimensional space and construct a classifier by searching for the hyperplane that gives the greatest separation between the classes.

We used our own implementation of the Vector Space Model and Weka implementations (Witten and Frank, 2005) of the other two algorithms.

4 Results

This system was applied to the NLM-WSD data set. Experiments were carried out using each of the three types of features (linguistic, CUI and MeSH) both alone and in combination. Ten-fold cross validation was used, and the figures we report are averaged across all ten runs.

Results from this experiment are shown in Table 1 which lists the performance using combinations of learning algorithm and features. The figure shown for each configuration represents the percentage of instances of ambiguous terms which are correctly disambiguated.

These results show that each of the three types of knowledge (linguistic, CUIs and MeSH) can be used to create a classifier which achieves a reasonable level of disambiguation since performance exceeds the relevant baseline score. This suggests that each of the knowledge sources can contribute to the disambiguation of ambiguous terms in biomedical text.

The best performance is obtained using a combination of the linguistic and MeSH features, a pattern observed across all test sets and machine learning algorithms. Although the increase in performance gained from using both the linguistic and MeSH features compared to only the linguistic features is modest it is statistically significant, as is the difference between using both linguistic and MeSH features compared with using the MeSH features alone (Wilcoxon Signed Ranks Test, $p < 0.01$).

Combining MeSH terms with other features generally improves performance, suggesting that the information contained in MeSH terms is distinct from the other knowledge sources. However, the inclusion of CUIs as features does not always improve performance and, in several cases, causes it to fall. This is consistent with McInnes et al. (2007) who concluded that CUIs were a useful information source for disambiguation of biomedical text but that they were not as robust as a linguistic knowledge source (unigrams) which they had used for a previous system. The most likely reason for this is that our approach relies on automatically assigned CUIs, provided by MetaMap, while the MeSH terms are assigned manually. We do not have access to a reliable assignment of CUIs to text; if we had WSD would not be necessary. On the other hand, reliably assigned MeSH terms are readily available in Medline. The CUIs assigned by MetaMap are noisy while the MeSH terms are more reliable and prove to be a more useful knowledge source for WSD.

The Vector Space Model learning algorithm performs significantly better than both Support Vector Machine and Naive Bayes (Wilcoxon Signed Ranks Test, $p < 0.01$). This pattern is observed regardless

Data sets	Features						
	Linguistic	CUI	MeSH	CUI+ MeSH	Linguistic +MeSH	Linguistic +CUI	Linguistic+ MeSH+CUI
	Vector space model						
All words	87.2	85.8	81.9	86.9	**87.8**	87.3	87.6
Joshi subset	82.3	79.6	76.6	81.4	**83.3**	82.4	82.6
Leroy subset	77.8	74.4	70.4	75.8	**79.0**	78.0	77.8
Liu subset	84.3	81.3	78.3	83.4	**85.1**	84.3	84.5
Common subset	79.6	75.1	70.4	76.9	**80.8**	79.6	79.2
	Naive Bayes						
All words	86.2	81.2	85.7	81.1	86.4	81.4	81.5
Joshi subset	80.6	73.4	80.1	73.3	80.9	73.7	73.8
Leroy subset	76.4	66.1	74.6	65.9	76.8	66.3	66.3
Liu subset	81.9	75.4	81.7	75.3	82.2	75.5	75.6
Common subset	76.7	66.1	74.7	65.8	77.2	65.9	65.9
	Support Vector Machine						
All words	85.6	83.5	85.3	84.5	86.1	85.3	85.6
Joshi subset	79.8	76.4	79.5	78.0	80.6	79.1	79.8
Leroy subset	75.1	69.7	72.6	72.0	76.3	74.2	74.9
Liu subset	81.3	78.2	81.0	80.0	82.0	80.6	81.2
Common subset	75.7	69.8	71.6	73.0	76.8	74.7	75.2

		Previous Approaches			
	MFS baseline	Liu et. al. (2004)	Leroy and Rindflesch (2005)	Joshi et. al. (2005)	McInnes et. al. (2007)
All words	78.0	–	–	–	85.3
Joshi subset	66.9	–	–	82.5	80.0
Leroy subset	55.3	–	65.5	77.4	74.5
Liu subset	69.9	78.0	–	84.9	82.0
Common subset	54.9	–	68.8	79.8	75.7

Table 1: Results from WSD system applied to various sections of the NLM-WSD data set using a variety of features and machine learning algorithms. Results from baseline and previously published approaches are included for comparison.

of which set of features are used, and it is consistent of the results in Senseval data from (Agirre and Martínez, 2004).

4.1 Per-Word Analysis

Table 2 shows the results of our best performing system (combination of linguistic and MeSH features using the Vector Space Model learning algorithm). Comparable results for previous supervised systems are also reported where available.[3] The MFS baseline for each term is shown in the leftmost column.

The performance of Leroy and Rindflesch's system is always lower than the best result for each word. The systems reported by Joshi et al. (2005) and McInnes et al. (2007) are better than, or the same as, all other systems for 14 and 12 words respectively. The system reported here achieves results equal to or better than previously reported systems for 33 terms.

There are seven terms for which the performance of our approach is actually lower than the MFS baseline (shown in *italics*) in Table 2. (In fact, the baseline outperforms all systems for four of these terms.) The performance of our system is within 1% of the baseline for five of these terms. The remaining pair, "blood pressure" and "failure", are included in the set of problematic words identified by (Weeber et al., 2001). Examination of the possible senses show that they include pairs with similar meanings. For

[3]It is not possible to directly compare our results with Liu et al. (2004) or Humphrey et al. (2006). The first report only optimal configuration for each term (combination of feature sets and learning algorithm) while the second do not assign senses to all of the instances of each ambiguous term (see Section 2).

	MFS baseline	Leroy and Rindflesch (2005)	Joshi et. al. (2005)	McInnes et. al. (2007)	Reported system
adjustment	62	57	71	70	**74**
association	100	-	-	97	**100**
blood pressure	54	46	**53**	46	46
cold	86	-	**90**	89	88
condition	90	-	-	**89**	**89**
culture	89	-	-	94	**95**
degree	63	68	89	79	**95**
depression	85	-	86	81	**88**
determination	79	-	-	81	**87**
discharge	74	-	95	**96**	95
energy	99	-	-	**99**	98
evaluation	50	57	69	73	**81**
extraction	82	-	84	**86**	85
failure	71	-	-	**73**	67
fat	71	-	**84**	77	**84**
fit	82	-	-	87	**88**
fluid	100	-	-	99	**100**
frequency	94	-	-	**94**	**94**
ganglion	93	-	-	94	**96**
glucose	91	-	-	90	**91**
growth	63	62	**71**	69	68
immunosuppression	59	61	**80**	75	**80**
implantation	81	-	**94**	92	93
inhibition	98	-	-	**98**	**98**
japanese	73	-	**77**	76	75
lead	71	-	89	90	**94**
man	58	80	89	80	**90**
mole	83	-	**95**	87	93
mosaic	52	66	**87**	75	**87**
nutrition	45	48	52	49	**54**
pathology	85	-	**85**	84	**85**
pressure	96	-	-	93	**95**
radiation	61	72	82	81	**84**
reduction	89	-	91	**92**	89
repair	52	81	87	**93**	88
resistance	97	-	-	96	**98**
scale	65	84	81	83	**88**
secretion	99	-	-	**99**	**99**
sensitivity	49	70	88	92	**93**
sex	80	-	**88**	87	87
single	99	-	-	98	**99**
strains	92	-	-	92	**93**
support	90	-	-	**91**	89
surgery	98	-	-	94	**97**
transient	99	-	-	98	**99**
transport	93	-	-	**93**	**93**
ultrasound	84	-	**92**	85	90
variation	80	-	-	91	**95**
weight	47	68	**83**	79	81
white	49	62	**79**	74	76

Table 2: Per-word performance of best reported systems.

example, the two senses which account for 98% of the instances of "blood pressure", which refer to the blood pressure within an organism and the result obtained from measuring this quantity, are very closely related semantically.

5 Conclusion

This paper has compared a variety of knowledge sources for WSD of ambiguous biomedical terms and reported results which exceed the performance of previously published approaches. We found that accurate results can be achieved using a combination of linguistic features commonly used for WSD

of general text and manually assigned MeSH terms. While CUIs are a useful source of information for disambiguation, they do not improve the performance of other features when used in combination with them. Our approach uses manually assigned MeSH terms while the CUIs are obtained automatically using MetaMap.

The linguistic knowledge sources used in this paper comprise a wide variety of features including n-grams and syntactic dependencies. We have not explored the effectiveness of these individually and this is a topic for further work.

In addition, our approach does not make use of the fact that MeSH terms are organised into a hierarchy. It would be interesting to discover whether this information could be used to improve WSD performance. Others have developed techniques to make use of hierarchical information in WordNet for WSD (see Budanitsky and Hirst (2006)) which could be adapted to MeSH.

References

E. Agirre and D. Martínez. 2004. The Basque Country University system: English and Basque tasks. In Rada Mihalcea and Phil Edmonds, editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 44–48, Barcelona, Spain, July.

A. Aronson, O. Bodenreider, H. Chang, S. Humphrey, J. Mork, S. Nelson, T. Rindflesch, and W. Wilbur. 2000. The NLM Indexing Initiative. In *Proceedings of the AMIA Symposium*.

A. Aronson. 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. In *Proceedings of the American Medical Informatics Association (AMIA)*, pages 17–21.

A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based measures of semantic distance. *Computational Linguistics*, 32(1):13–47.

S. Humphrey, W. Rogers, H. Kilicoglu, D. Demner-Fushman, and T. Rindflesch. 2006. Word Sense Disambiguation by selecting the best semantic type based on Journal Descriptor Indexing: Preliminary experiment. *Journal of the American Society for Information Science and Technology*, 57(5):96–113.

L. Humphreys, D. Lindberg, H. Schoolman, and G. Barnett. 1998. The Unified Medical Language System: An Informatics Research Collaboration. *Journal of the American Medical Informatics Association*, 1(5):1–11.

M. Joshi, T. Pedersen, and R. Maclin. 2005. A Comparative Study of Support Vector Machines Applied to the Word Sense Disambiguation Problem for the Medical Domain. In *Proceedings of the Second Indian Conference on Artificial Intelligence (IICAI-05)*, pages 3449–3468, Pune, India.

G. Leroy and T. Rindflesch. 2005. Effects of Information and Machine Learning algorithms on Word Sense Disambiguation with small datasets. *International Journal of Medical Informatics*, 74(7-8):573–585.

H. Liu, V. Teller, and C. Friedman. 2004. A Multi-aspect Comparison Study of Supervised Word Sense Disambiguation. *Journal of the American Medical Informatics Association*, 11(4):320–331.

D. McCarthy, R. Koeling, J. Weeds, and J. Carroll. 2004. Finding predominant senses in untagged text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Lingusitics (ACL-2004)*, pages 280–287, Barcelona, Spain.

B. McInnes, T. Pedersen, and J. Carlis. 2007. Using UMLS Concept Unique Identifiers (CUIs) for Word Sense Disambiguation in the Biomedical Domain. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pages 533–537, Chicago, IL.

R. Mihalcea, T. Chklovski, and A. Kilgarriff. 2004. The Senseval-3 English lexical sample task. In *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain.

S. Nelson, T. Powell, and B. Humphreys. 2002. The Unified Medical Language System (UMLS) Project. In Allen Kent and Carolyn M. Hall, editors, *Encyclopedia of Library and Information Science*. Marcel Dekker, Inc.

T. Pedersen. 2001. A Decision Tree of Bigrams is an Accurate Predictor of Word Sense. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01)*, pages 79–86, Pittsburgh, PA., June.

A. Ratnaparkhi. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142.

M. Stevenson and Y. Wilks. 2001. The Interaction of Knowledge Sources in Word Sense Disambiguation. *Computational Linguistics*, 27(3):321–350.

M. Weeber, J. Mork, and A. Aronson. 2001. Developing a Test Collection for Biomedical Word Sense Disambiguation. In *Proceedings of AMAI Symposium*, pages 746–50, Washington, DC.

I. Witten and E. Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco.

Automatic inference of indexing rules for MEDLINE

Aurélie Névéol and Sonya E. Shooshan
National Library of Medicine
8600 Rockville Pike
Bethesda, MD 20894, USA
{neveola,sonya}@nlm.nih.gov

Vincent Claveau
IRISA - CNRS
Campus de Beaulieu
35042 Rennes, France
Vincent.Claveau@irisa.fr

Abstract

This paper describes the use and customization of Inductive Logic Programming (ILP) to infer indexing rules from MEDLINE citations. Preliminary results suggest this method may enhance the subheading attachment module of the Medical Text Indexer, a system for assisting MEDLINE indexers.

1 Introduction

Indexing is a crucial step in any information retrieval system. In MEDLINE®, a widely used database of the biomedical literature, the indexing process involves the selection of Medical Subject Headings (MeSH®) in order to describe the subject matter of articles. The need for automatic tools to assist human indexers in this task is growing with the increasing number of publications in MEDLINE. The Medical Text Indexer (MTI) (Aronson et al., 2004) has been available at the U.S. National Library of Medicine (NLM) since 2002 to provide indexers with MeSH main heading recommendations (e.g. *Aphasia*, *Patient Care*...) when they create MEDLINE citations. This paper describes a method to enhance MTI with the capacity to attach appropriate MeSH subheadings (e.g. *metabolism*, *pharmacology*) to these main headings in order to provide MeSH pair recommendations (e.g. *aphasia/metabolism*), which are more specific and therefore a significant asset to NLM indexers.

Subheading attachment can be accomplished using indexing rules such as:

```
If a main heading from the "Anatomy"
tree and a "Carboxylic Acids" term are
recommended for indexing, then the pair
"[Carboxylic Acids]/pharmacology" should
also be recommended.
```

Sets of manual rules developed for a few subheadings show good precision but low recall. The development of new rules is a complex, time-consuming task. We investigate a novel approach adapting Inductive Logic Programming (ILP) to the context of MEDLINE, which requires efficient processing of large amounts of data.

2 Use of Inductive Logic Programming

ILP is a supervised machine learning technique used to infer rules that are expressed with logical clauses (Prolog clauses) based on a set of examples also represented using Prolog. A comprehensive description of ILP can be found in (Muggleton and Raedt, 1994). We selected this method because it is able to provide simple representations for relational problems and produces rules that can be easily interpreted. One caveat to the use of ILP is the complexity of rule inference from large sets of positive and negative examples. Considering each of the 24,000 MeSH main headings independently would not be computationally feasible. For this reason, based on work by Buntine (1988) we introduce a new definition of subsumption that allows us to go through the set of examples efficiently by exploiting hierarchical relationships between main headings. This type of subsumption is in fact suitable for any rule inference problem involving structured knowledge encoded by ontologies.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 88–89,
Columbus, Ohio, USA, June 2008.

Subheading	Method	Nb. rules	Precision (%)	Recall(%)	F-measure(%)
Overall	ILP	587	47	**32**	**38**
	Manual	69	**59**	10	18
	Baseline	-	32	11	16

Table 1: Performance on the test corpus using MTI main heading recommendations

3 Experiments

ILP rules were induced using a training corpus of 100,000 citations randomly chosen from MEDLINE 2006. Another corpus of 100,000 MEDLINE 2006 citations was used for testing. ILP rules were applied on the test corpus using main headings automatically retrieved by MTI as triggers. The performance of ILP was compared to manual rules and a baseline consisting of randomly formed pairs according to their distribution in MEDLINE prior to 2006. Overall results obtained on 4 subheadings are presented in Table 1.

4 Discussion

Performance. As expected, the use of MTI to produce main heading recommendations used as triggers for the rules results in comparable precision but lower recall compared to the theoretical assessment. In spite of this, the performance obtained by ILP rules is superior to the baseline and shows the best F-measure. The precision obtained by the manual rules, when they exist, is higher, but they produce a recall inferior to ILP and even to the baseline method.

ILP *vs.* manual rules. A detailed analysis of the rules obtained shows that not all ILP rules are easily understood by indexers. This is due to some unexpected regularities which do not seem to be relevant but nonetheless achieved good results on the training data used to infer rules.

Furthermore, we noticed that while most rules typically contain a "trigger term" (e.g. *Anatomy* in our previous example) and a "target term" (e.g. *Carboxylic Acids* above), in some ILP rules the target term can also serve as the trigger term. Some changes in the ILP inferring process are foreseen in order to prevent the production of such rules.

Rule filtering *vs.* manual review. Preliminary experiments with producing ILP rules suggested that improvement could be achieved by 1/ filtering out rules that showed a comparatively low precision on the training corpus when applied to main headings retrieved by MTI; and 2/ by having an indexing expert review the rules to improve their readability. On most subheadings, filtering had little impact but generally tended to improve precision while F-measure stayed the same, which was our goal. The manual review of the rules seemed to degrade the performance obtained with the original ILP.

5 Conclusion and perspectives

We have shown that ILP is an adequate method for automatically inferring indexing rules for MEDLINE. Further work will be necessary in order to obtain rules for all 83 MeSH subheadings. Subsequently, the combination of ILP rules with other subheading attachment methods will be assessed. We anticipate that the rule sets we have obtained will be integrated into MTI's subheading attachment module.

Acknowledgments

This study was supported in part by the Intramural Research Programs of NIH, NLM. A. Névéol was supported by an appointment to the NLM Research Participation Program administered by ORISE through an inter-agency agreement between the U.S. Department of Energy and NLM.

References

Alan R. Aronson, James G. Mork, Clifford W. Gay, Susanne M. Humphrey, and Willie J. Rogers. 2004. The NLM Indexing Initiative's Medical Text Indexer. In *Proceedings of Medinfo 2004*, San Francisco, California, USA.

Wray L. Buntine. 1988. Generalized Subsumption and its Application to Induction and Redundancy. *Artificial Intelligence*, 36:375–399.

Stephen Muggleton and Luc De Raedt. 1994. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679.

Prediction of Protein Sub-cellular Localization using Information from Texts and Sequences

Hong-Woo Chun[1,2,3] **Chisato Yamasaki**[2,3] **Naomi Saichi**[2,3] **Masayuki Tanaka**[2,3]
chun@dbcls.rois.ac.jp, {chisato-yamasaki, nao-saichi, masa-tanaka}@aist.go.jp
Teruyoshi Hishiki[3] **Tadashi Imanishi**[3,5] **Takashi Gojobori**[3,6]
{t-hishiki, t.imanishi, t-gojobori}@aist.go.jp
Jin-Dong Kim[4] **Jun'ichi Tsujii**[4,7,8] **Toshihisa Takagi**[1,9]
{jdkim, tsujii}@is.s.u-tokyo.ac.jp, takagi@dbcls.rois.ac.jp
[1] Database Center for Life Science, Research Organization of Information and System, Engineering 12th Bldg., University of Tokyo, 2-11-16 Yayoi, Bunkyo-ku, Tokyo, 113-0032, Japan
[2] Japan Biological Information Research Center, Japan Biological Informatics Consortium
[3] Biological Information Research Center, National Institute of Advanced Industrial Science and Technology, Japan
[4] Department of Computer Science, University of Tokyo, Japan
[5] Graduate School of Information Science and Technology, Hokkaido University, Japan
[6] Center for Information Biology and DNA Data Bank of Japan, National Institute of Genetics
[7] School of Informatics, University of Manchester, UK
[8] National Centre for Text Mining, UK
[9] Department of Computational Biology, University of Tokyo, Japan

Abstract

This paper presents a novel prediction approach for protein sub-cellular localization. We have incorporated text and sequence-based approaches.

1 Introduction

Natural Language Processing (NLP) has tackled and solved a lot of prediction problems in Biology. One practical research issue is *Protein Sub-Cellular Localization (PSL) Prediction*. Many previous approaches have combined information from both texts and sequences by a machine learning (ML) technique (Shatkay et al., 2007). All of them have not used traditional NLP techniques such as parsing. Our aim is to develop a novel PSL prediction system using information from texts and sequences. At the same time, we demonstrated the effectiveness of the traditional NLP and the sequence-based features in the viewpoint of the text-based approach.

2 Methodology

A Maximum Entropy-based ML technique has been used to combine information from both texts and sequences. To develop a supervised ML-based prediction system, an annotated corpus is needed to train the system. However, there is no publicly available corpus that contains the PSL. Therefore, we have constructed a corpus using GENIA corpus as an initial data, because the annotation of *Protein* and *Cellular component* in GENIA corpus is already done by human experts. The new types of annotation contain two tasks. The first annotation is to classify 1,117 cellular components in GENIA corpus into 11 locations, and the second annotation is to categorize a relation between a protein and a location into positive, negative, and neutral. Biologists selected 11 locations based on *Gene Ontology*: *Cytoplasm, Cytoskeleton, Endoplasmic reticulum, Extracellular, Golgi apparatus, Granule, Lysosome, Mitochondria, Nucleus, Peroxisome,* and *Plasma membrane*. The number of co-occurrences in GENIA corpus is 864. [1] Three human experts annotated with 79.49% of inter-annotator agreement. For calculating the inter-annotator agreement, all annotators annotated 117

[1] The co-occurrence in the proposed approach is a sentence that contains at least one pair of protein and cellular component names.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 90–91,
Columbus, Ohio, USA, June 2008.

Location	# Relevant relations	Performance : F-score (Precision, Recall)			
		Baseline	Text	Sequence	Text + Sequence
Nucleus	173	0.282 (0.164, 1.0)	0.764 (0.736, 0.794)	0.725 (0.569, 1.000)	**0.778 (0.758, 0.798)**
Cytoplasm	94	0.163 (0.089, 1.0)	**0.828 (0.804, 0.852)**	0.788 (0.657, 0.984)	**0.828 (0.804, 0.852)**
Plasma membrane	23	0.043 (0.022, 1.0)	0.875 (0.814, 0.946)	0.857 (0.766, 0.973)	**0.885 (0.841, 0.932)**

Table 1: Performance of protein sub-cellular localization prediction for each location.

co-occurrences. From the texts, we used eight features: (1) protein and cellular component names annotated by human experts, (2) adjacent one and two words of names, (3) bag of words, (4) order of names, (5) distance between names, (6) syntactic category of names, (7) predicates of names, and (8) part-of-speech of predicates. To analyze the syntactic structure, we used the ENJU full parser whose output is predicate-argument structures of a sentence.

To combine the information from sequences, we attempted to predict PSL for all proteins in GENIA corpus by two existing sequence-based methods: WoLF PSORT (Horton et al., 2006) and SOSUI (Hirokawa et al., 1998). Approximately 14% of protein names in GENIA corpus obtained results. From the sequences, we used two features: (1) existence of the sequence-based results, and (2) the number of sequence-based results.

3 Experimental results and Conclusion

The proposed approach has integrated text and sequence-based approaches. To evaluate the system, we performed 10-fold cross validation using 864 co-occurrences including positive, negative, and neutral relations. We measured the precision, recall, and F-score of the system for all experiments. Among 864 co-occurrences in GENIA corpus, 301 positive or negative co-occurrences have been considered as relevant relations, and the remaining 563 neutral relations have been considered as irrelevant relations.

Four approaches have been compared based on three locations in Table 1. The four approaches are *baseline, text-based approach, sequence-based approach, and integration of the text and sequence-based approaches*. Baseline experiment used an assumption: *there is a relevant relation if a protein and a cellular component names occur together in a co-occurrence*. The three locations selected when there are the sequence-based results and the number of relevant relations is more than *one*. All experiments showed that the integration of text and sequence-based approaches is the best, even though the experiments for *Cytoplasm* showed the best performance at both the text-based approach and the integration approach.

A new prediction method has been developed for protein sub-cellular localization, and it has integrated text and sequence-based approach using an ML technique. The traditional NLP techniques contributed to improve performance of the text-based approach, and the text and sequence-based approaches reciprocally contributed to obtain a improved PSL prediction method. The newly constructed corpus will be included in the next version of GENIA corpus. There are weak points in the proposed approach. The current evaluation method has been focusing on evaluating the text-based approach, and the results of the sequence-based approach were obtained for only 14% of proteins in GENIA corpus, so these situations might be the reason that the sequence-based approach did contribute a little. Thus, we need to evaluate the proposed approach with a more reasonable method.

Acknowledgments

We acknowledge Fusano Todokoro for her technical assistance.

References

Paul Horton, Keun-Joon Park, Takeshi Obayashi and Kenta Nakai. 2006. *Protein Subcellular Localization Prediction with WoLF PSORT. Asia Pacific Bioinformatics Conference (APBC)*, pp. 39–48.

Takatsugu Hirokawa, Seah Boon-Chieng and Shigeki Mitaku. 1998. *SOSUI: classification and secondary structure prediction system for membrane proteins. Bioinformatics*, 14(4): pp. 378–379.

Hagit Shatkay, Annette Höglund, Scott Brady, Torsten Blum, Pierre Dönnes and Oliver Kohlbacher. 2007. *SherLoc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. Bioinformatics.*, 23(11): pp. 1410–1417

A Pilot Annotation to Investigate Discourse Connectivity in Biomedical Text

Hong Yu, Nadya Frid, Susan McRoy
University of Wisconsin-Milwaukee
P.O.Box 413
Milwaukee, WI 53201
Hongyu,frid,mcroy@uwm.edu

Rashmi Prasad, Alan Lee, Aravind Joshi
University of Pennsylvania
3401 Walnut Street
Philadelphia, PA 19104, USA
Rjprasad,aleewk,joshi@seas.upenn.edu

Abstract

The goal of the Penn Discourse Treebank (PDTB) project is to develop a large-scale corpus, annotated with coherence relations marked by discourse connectives. Currently, the primary application of the PDTB annotation has been to news articles. In this study, we tested whether the PDTB guidelines can be adapted to a different genre. We annotated discourse connectives and their arguments in one 4,937-token full-text biomedical article. Two linguist annotators showed an agreement of 85% after simple conventions were added. For the remaining 15% cases, we found that biomedical domain-specific knowledge is needed to capture the linguistic cues that can be used to resolve inter-annotator disagreement. We found that the two annotators were able to reach an agreement after discussion. Thus our experiments suggest that the PDTB annotation can be adapted to new domains by minimally adjusting the guidelines and by adding some further domain-specific linguistic cues.

1 Introduction

Large scale annotated corpora, e.g., the Penn TreeBank (PTB) project (Marcus et al. 1993), have played an important role in text-mining. The Penn Discourse Treebank (PDTB) (http://www.seas.upenn.edu/~pdtb) (Prasad et al. 2008a) annotates the *argument structure*, *semantics*, and *attribution* of discourse connectives and their arguments. The current release of PDTB-2.0 contains the annotations of 1,808 Wall Street Journal articles (~1 million words) from the Penn TreeBank (Marcus et al. 1993) II distribution and a total of 40,600 discourse connective tokens (Prasad et al. 2008b). This work examines whether the PDTB annotation guidelines can be adapted to a different genre, the biomedical literature.

2 Notation

A discourse connective can be defined as a word or multiword expression that signals a discourse relation. Discourse connectives can be subordinating conjunctions (e.g., *because, when, although*), coordinating conjunctions (e.g., *but, or, nor*) and adverbials (e.g., *however, as a result, for example*). A discourse connective takes in two arguments, *Arg1* and *Arg2*. Arg2 is the argument that appears in the clause that is syntactically bound to the connective and Arg1 is the other argument. In the sentence "*John failed the exam* because **he was lazy**" the discourse connective is underlined, Arg1 appears in italics and Arg2 appears in bold.

3 A Pilot Annotation

Following the PDTB annotation manual (Prasad et al. 2008b), we conducted a pilot annotation of discourse connectivity in biomedical text. As an initial step, we only annotated the three most

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 92–93,
Columbus, Ohio, USA, June 2008.

important components of a discourse relation; namely, a discourse connective and its two arguments; we did not annotate attribution. Two linguist annotators independently annotated one full-text biomedical article (Verpy et al. 1999) that we randomly selected. The article is 4,937 tokens long. When the annotation work was completed, we measured the inter-annotator agreement, following the PDTB exact match criterion (Miltsakaki et al. 2004). According to this criterion, a discourse relation is in disagreement if there is disagreement on any text-span (i.e., the discourse connective or any of its two arguments). In addition, we also measured the agreement in the components (i.e., discourse connectives and the arguments). We discussed the annotation results and made suggestions to adapt the PDTB guidelines to biomedical text.

4 Results and Discussion

The first annotator identified 74 discourse connectives, and the second annotator identified 75, 68 of which were the same as those identified by the first annotator. The combined total number of discourse connectives was 81. The overall agreement in discourse connective identification was 68/81=84%.

Of the 68 discourse connectives that were annotated by both annotators, 31 were an exact match, 31 had an exact match for Arg1, and 54 had an exact match for Arg2. The overall agreement for the 68 discourse relations is 45.6% for exact match, 45.6% for Arg1, and 79.4% for Arg2. The PDTB also reported a higher level of agreement in annotating Arg2 than in annotating Arg1 (Miltsakaki et al. 2004). We manually analyzed the cases with disagreement. We found the disagreements are nearly all related to the annotation of citation references, supplementary clauses, and other conventions. When a few conventions for these cases were added, the inter-annotator agreement went up to 85%. We also found that different interpretation of a relation and its arguments by annotators plays an important role for the remaining 15% inconsistency, and domain-specific knowledge is necessary to resolve such cases.

5 New Conventions

After the completion of the pilot annotation and the discussion, we decided to add the following conventions to the PDTB annotation guidelines to address the characteristics of biomedical text:

i. Citation references are to be annotated as a part of an argument because the inclusion will benefit many text-mining tasks including identifying the semantic relations among citations.
ii. Clausal supplements (e.g., relative or parenthetical constructions) that modify arguments but are not minimally necessary for the interpretation of the relation, are annotated as part of the arguments.
iii. We will annotate a wider variety of nominalizations as arguments than allowed by the PDTB guidelines.

We anticipate that these changes will both decrease the amount of effort required for annotation and increase the reliability of the annotation.

6 References

Marcus M, Santorini B, Marcinkiewicz M (1993) Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics 19

Miltsakaki E, Prasad R, Joshi A, Webber B (2004) Annotating discourse connectives and their arguments. Paper presented at Proceedings of the NAACL/HLT Workshop: Frontiers in Corpus Annotation

Prasad R, Dinesh N, Lee A, Miltsakaki E, Robaldo L, Joshi A, Webber B (2008a) The Penn Discourse Treebank 2.0. Paper presented at The 6th International Conference on Language Resources and Evaluation (LREC). Marrakech, Morroco

Prasad R, Miltsakaki E, Dinesh N, Lee A, Joshi A, Robaldo L, Webber B (2008b) The Penn Discourse TreeBank 2.0 Annotation Manual. Technical Report: IRCS-08-01

Verpy E, Leibovici M, Petit C (1999) Characterization of otoconin-95, the major protein of murine otoconia, provides insights into the formation of these inner ear biominerals. Proc Natl Acad Sci U S A 96:529-534

Conditional Random Fields and Support Vector Machines for Disorder Named Entity Recognition in Clinical Texts

Dingcheng Li
University of Minnesota
Minneapolis, Minnesota, USA
lixxx345@umn.edu

Karin Kipper-Schuler
Mayo Clinic College of Medicine
Rochester, Minnesota, USA
schuler.karin@mayo.edu

Guergana Savova
Mayo Clinic College of Medicine
Rochester, Minnesota, USA
savova.guergana@mayo.edu

Abstract

We present a comparative study between two machine learning methods, Conditional Random Fields and Support Vector Machines for clinical named entity recognition. We explore their applicability to clinical domain. Evaluation against a set of gold standard named entities shows that CRFs outperform SVMs. The best F-score with CRFs is 0.86 and for the SVMs is 0.64 as compared to a baseline of 0.60.

1 Introduction and background

Named entity recognition (NER) is the discovery of named entities (NEs), or textual mentions that belong to the same semantic class. In the biomedical domain NEs are diseases, signs/symptoms, anatomical signs, and drugs. NER performance is high as applied to scholarly text and newswire narratives (Leaman et al., 2008). Clinical free-text, on the other hand, exhibits characteristics of both informal and formal linguistic styles which, in turn, poses challenges for clinical NER. Conditional Random Fields (CRFs) (Lafferty et al., 2001) and and Support Vector Machines (SVMs) (Cortes and Vapnik, 1995) are machine learning techniques which can handle multiple features during learning. CRFs' main strength lies in their ability to include various unrelated features, while SVMs' in the inclusion of overlapping features. Our goal is to compare CRFs and SVMs performance for clinical NER with focus on disease/disorder NEs.

2 Dataset and features

Our dataset is a gold standard corpus of 1557 single- and multi-word disorder annotations (Ogren et al., 2008). For training and testing the CRF and SVM models the IOB (inside-outside-begin) notation (Leaman, 2008) was applied. In our project, we used 1265 gold standard annotations for training and 292 for testing. The features used for the learning process are described as follows. *Dictionary look-up* is a binary value feature that represents if the NE is in the dictionary (SNOMED-CT). *Bag of Words (BOW)* is a representation of the context by the unique words in it. *Part-of-speech tags (POS)* of BOW is the pos tags of the context words. *Window size* is the number of tokens representing context surrounding the target word. *Orientation(*left or right) is the location of the feature in regard to the target word. *Distance* is the proximity of the feature in regard to the target word *Capitalization* has one of the four token-based values: all upper case, all lower case, mixed_case and initial upper case. *Number features* refer to the presence or absence of related numbers. Feature sets are in Table 1.

3 Results and discussion

Figure 1 shows the CRF results. The F-scores, recall and precision for the baseline dictionary look-up are 0.604, 0.468 and 0.852 respectively. When BOW is applied in feature combination 2 results improve sharply adding 0.15, 0.17 and 0.08 points respectively. The F-score, recall and precision improve even further with the capitalization feature to 0.858, 0.774 and 0.963 respectively. Figure 2 shows SVM results. The addition of more features to the model did not show an upward trend. The best results are with feature combination 1 and 3. The F-score reaches 0.643, which although an improvement over the baseline greatly underperforms CRF results. BOW features seem not discriminative with SVMs. When the window size increases to 5, performance decreases as demonstrated in feature combinations 2, 4 and 8. Results with feature combination 4, in particular, has a pronounced downward trend. Its F-score is 0.612, a decrease by 0.031 compared with Test 1 or Test 3. Its recall and precision are 0.487 and 0.822 respectively, a decrease by 0.036 and 0.01 respectively. This supports the results achieved with CRFs where a smaller window size yields better performance.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 94–95,
Columbus, Ohio, USA, June 2008.

No	*Features*
1	dictionary look-up (baseline)
2	dictionary look-up+BOW+Orientation+distance (Window 5)
3	dictionary look-up + BOW + Orientation + distance (Window 3)
4	dictionary look-up + BOW + POS + Orientation + distance (Window 5)
5	dictionary look-up + BOW +POS + Orientation + distance (Window 3)
6	dictionary look-up + BOW +POS + Orientation + distance (Window 3) + bullet number
7	dictionary look-up + BOW + POS + Orientation + distance(Window 3) + measurement
8	dictionary look-up + BOW + POS + Orientation + distance (Window 5) + neighboring number
9	dictionary look-up + BOW +POS + Orientation + distance (Window 3) + neighboring number
10	dictionary look-up + BOW +POS + Orientation + distance (Window 3)+neighboring number+measurement
11	dictionary look-up+BOW+POS+Orientation (Window 3)+neighboring number+bullet number + measurement
12	dictionary look-up + BOW +POS + Orientation +distance (Window 3) + neighboring number + bullet number + measurement + capitalization

Table 1: Feature combinations

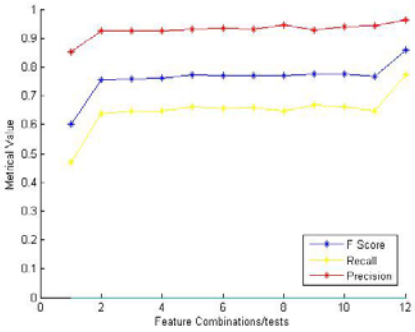

Figure 1: CRF evaluation results

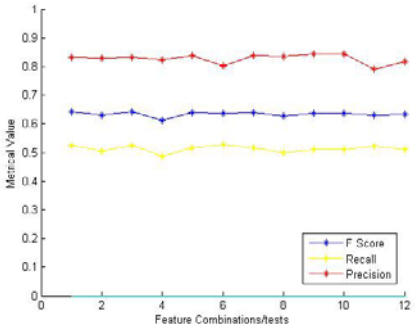

Figure 2: SVM evaluation results

As the results show, context represented by the BOW feature plays an important role indicating the importance of the words surrounding NEs. On the other hand, POS tag features did not bring much improvement, which perhaps hints at a hypothesis that grammatical roles are not as important as context in clinical text. Thirdly, a small window size is more discriminative. Clinical notes are unstructured free text with short sentences. If a larger window size is used, many words will share similar features. Fourthly, capitalization is highly discriminative. Fifthly, as a finite state machine derived from HMMs, CRFs can naturally consider state-to-state dependences and feature-to-state dependences. On the other hand, SVMs do not consider such dependencies. SVMs separate the data into categories via a kernel function. They implement this by mapping the data points onto an optimal linear separating hyperplane. Finally, SVMs do not behave well for large number of feature values. For large number of feature values, it would be more difficult to find discriminative lines to categorize the labels.

4 Conclusion and future work

We investigated the use of CRFs and SVMs for disorder NER in clinical free-text. Our results show that, in general, CRFs outperformed SVMs. We demonstrated that well-chosen features along with dictionary-based features tend to improve the CRF model's performance but not the SVM's.

Acknowledgements

The work was partially supported by a Biomedical Informatics and Computational Biology scholarship from the University of Minnesota.

References

Corinna Cortes and Vladimir Vapnik. Support-vector network. *Machine Learning*, 20:273-297, 1995.

John Lafferty, Andrew McCallum and Fernando Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning* (ICML-2001), 2001.

Robert Leaman and Graciela Gonzalez. BANNER: an Executable Survey of Advances in Biomedical Named Entity Recognition. *Pacific Symposium on Biocomputing* 13:652-663. 2008.

Philip Ogren, Guergana Savova and Christopher G Chute. Constructing evaluation corpora for automated clinical named entity recognition. *Proc LREC* 2008.

Using Natural Language Processing to Classify Suicide Notes

John P. Pestian*, Pawel Matykiewicz, Jacqueline Grupp-Phelan,
Sarah Arszman Lavanier, Jennifer Combs, and Robert Kowatch
Cincinnati Children's Hospital Medical Center
Cincinnati, OH 45220, USA
john.pestian@cchmc.org

Abstract

We hypothesize that machine-learning algorithms (MLA) can classify completer and simulated suicide notes as well as mental health professionals (MHP). Five MHPs classified 66 simulated or completer notes; MLAs were used for the same task. Results: MHPs were accurate 71% of the time; using the sequential minimization optimization algorithm (SMO) MLAs were accurate 78% of the time. There was no significant difference between the MLA and MPH classifiers. This is an important first step in developing an evidence based suicide predictor for emergency department use.

1 Problem

Suicide is the third leading cause of death in adolescents and a leading cause of death in the United States[1]. Those who attempt suicide usually arrive at the Emergency Department seeking help. These individuals are at risk for a repeated attempt, that may lead to a completed suicide[2]. We know of no evidence-based risk assessment tool for predicting repeated suicide attempts. Thus, Emergency Medicine clinicians are often left to manage suicidal patients by clinical judgment alone. This research focuses on the initial stage for constructing such an evidence based tool, the Psychache[3] Index. Our efforts herein posit that suicide notes are an artifact of a victim's thoughts and that the thoughts between completers and attempters are different. Using natural language processing we attempt to distinguish between completer notes and notes that have been simulated by individuals who match the profile of the completer. Understanding how to optimize classification methods between these types of notes prepares us for future work that can include clinical and biological factors.

2 Methods

Suicidal patients are classified into three categories: ideators —those who think about committing suicide, attempters —those who attempt suicide, and completers —those who complete suicide. This research focuses on the completers and a group of individuals called *simulators*. These simulators were matched to completers by age, gender and socioeconomic status and asked to write a suicide note[4]. Suicide notes from 33 completers and 33 simulators were annotated with linguistic characteristics using a perl-program with the EN:Lingua:Tagger module. Emotional characteristics were annotated by assigning terms in the note to a suicide-emotion ontology that was developed from a meta analysis of 2,166 suicide related manuscripts and validated with expert opinion. This ontology includes such classes as: affection, anger, depression, and worthlessness. Each class had multiple concepts, i.e, affection→ love, concern for others, and gratitude. Three MHPs read each note and tagged emotion-words found in the notes with the appropriate classes and concepts. Analysis of variance between structures was conducted to insure that there actually was a difference that could be detected. Emotional annotations were used for machine-learning.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 96–97,
Columbus, Ohio, USA, June 2008.

We then tested the hypothesis that MLAs could distinguish between completer and simulated notes as well as MHPs. Copies of the notes were given to five MHPs who classified them as either written by a completer or an simulator. MLA feature space was defined by matrix of selected characteristics from four sources: words, parts of speech, concepts, and readability indexes. Collinearity was eliminated by removing highly correlated features. The final feature space included: specific words (such as "love", "life", "no"), specific parts of speech (such as, personal pronouns, verbs) Kincaid readability index and emotional concepts (such as anger, and hopelessness). We then tested the following algorithms' ability to distinguish between completer and simulator notes: *decision trees* - J48, C4.5, LMT, DecisionStump, M5P; *classification rules* - JRip, M5, OneR, PART; *function models* - SMO, logistic builds, multinomial logistic regression, linear regression; *lazy learners* and *meta learners*[5].

3 Results

A significant difference was found between the linguistic and emotional characteristics of the notes. Linguistic differences (completer/simulated): word count 120/66 p=0.007, verbs 25/13 p=0.012, nouns 28/12 p=0.0001, and prepositions 20/10 p=0.005. This difference justified testing the classification hypothesis. Emotionally, completers gave away their possessions 20% of the time, simulators, never did. Mental health experts accurately classified the notes 71% of the time. The MLAs were accurate 60-79% of the time with SMO giving the highest results when the word count, part-of-speech, and readability vectors were included. Performance weakened when the emotional vector was included, yet the emotional vector was the primary source of data for the MHPs.

4 Conclusion

Machine learning methods for classifying suicide and non-suicide notes are promising. Future efforts to represent the thoughts of the suicidal patient will require larger sample sizes, inclusion of attempters response to open-ended questions, biological and clinical characteristics.

5 Acknowledgements

We acknowledge Drs. AA Leenaars, ES Shneidman, the divisions of Biomedical Informatics, Emergency Medicine and Psychiatry at Cincinnati Children's Hospital Medical Center, University of Cincinnati and Ohio Third Frontier program for their generous support of this work.

References:

[1] Jeffrey A Bridge, Tina R Goldstein, and David A Brent. Adolescent suicide and suicidal behavior. *J Child Psychol Psychiatry*, 47(3-4):372–394, 2006.

[2] P M Lewinsohn, P Rohde, and J R Seeley. Psychosocial risk factors for future adolescent suicide attempts. *J Consult Clin Psychol*, 62(2):297–305, 1994.

[3] E S Shneidman. Suicide as psychache. *J Nerv Ment Dis*, 181(3):145–147, 1993.

[4] ES Shneidman and NL Farberow. *Clues to Suicide*. McGraw Hill Paperbacks, 1957.

[5] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools ad Techniques*. Morgan Kaufman, 2nd edition, 2005.

Extracting Protein-Protein Interaction based on Discriminative Training of the Hidden Vector State Model

Deyu Zhou and Yulan He
Informatics Research Centre, The University of Reading, Reading RG6 6BX, UK
Email:d.zhou@reading.ac.uk, y.he@reading.ac.uk

1 Introduction

The knowledge about gene clusters and protein interactions is important for biological researchers to unveil the mechanism of life. However, large quantity of the knowledge often hides in the literature, such as journal articles, reports, books and so on. Many approaches focusing on extracting information from unstructured text, such as pattern matching, shallow and deep parsing, have been proposed especially for extracting protein-protein interactions (Zhou and He, 2008).

A semantic parser based on the Hidden Vector State (HVS) model for extracting protein-protein interactions is presented in (Zhou et al., 2008). The HVS model is an extension of the basic discrete Markov model in which context is encoded as a stack-oriented state vector. Maximum Likelihood estimation (MLE) is used to derive the parameters of the HVS model. In this paper, we propose a discriminative approach based on parse error measure to train the HVS model. To adjust the HVS model to achieve minimum parse error rate, the generalized probabilistic descent (GPD) algorithm (Kuo et al., 2002) is used. Experiments have been conducted on the GENIA corpus. The results demonstrate modest improvements when the discriminatively trained HVS model outperforms its MLE trained counterpart by 2.5% in F-measure on the GENIA corpus.

2 Methodologies

The Hidden Vector State (HVS) model (He and Young, 2005) is a discrete Hidden Markov Model (HMM) in which each HMM state represents the state of a push-down automaton with a finite stack size.

Normally, MLE is used for generative probability model training in which only the correct model needs to be updated during training. It is believed that improvement can be achieved by training the generative model based on a discriminative optimization criteria (Klein and Manning, 2002) in which the training procedure is designed to maximize the conditional probability of the parses given the sentences in the training corpus. That is, not only the likelihood for the correct model should be increased but also the likelihood for the incorrect models should be decreased.

Assuming the most likely semantic parse tree $\hat{C} = C_j$ and there are altogether M semantic parse hypotheses for a particular sentence W, a parse error measure (Juang et al., 1993; Chou et al., 1993; Chen and Soong, 1994) can be defined as

$$d(W) = -\log P(W, C_j) + \log[\frac{1}{M-1} \sum_{i, i \neq j} P(W, C_i)^{\eta}]^{\frac{1}{\eta}} \quad (1)$$

where η is a positive number and is used to select competing semantic parses. When $\eta = 1$, the competing semantic parse term is the average of all the competing semantic parse scores. When $\eta \rightarrow \infty$, the competing semantic parse term becomes $\max_{i.i \neq j} P(W, C_i)$ which is the score for the top competing semantic parse. By varying the value of η, we can take all the competing semantic parses into consideration. $d(W) > 0$ implies classification error and $d(W) \leq 0$ implies correct decision.

The sigmoid function can be used to normalize $d(W)$ in a smooth zero-one range and the loss function is thus defined as (Juang et al., 1993):

$$\ell(W) = \text{sigmoid}(d(W)) \quad (2)$$

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 98–99,
Columbus, Ohio, USA, June 2008.

where

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-\gamma x}} \quad (3)$$

Here, γ is a constant which controls the slope of the sigmoid function.

The update formula is given by:

$$\lambda^{k+1} = \lambda^k - \epsilon^k \nabla \ell(W_i, \lambda^k) \quad (4)$$

where ϵ^k is the step size.

Using the definition of $\ell(W_i, \lambda^k)$ and after working out the mathematics, we get the update formulae 5, 6, 7,

$$\left(\log P(n|\mathbf{c}')\right)^* = \log P(n|\mathbf{c}') - \epsilon\gamma\ell(d_i)(1 - \ell(d_i)) \left(-I(C_j, n, \mathbf{c}') + \sum_{i, i \neq j} I(C_i, n, \mathbf{c}') \frac{P(W_i, C_i, \lambda)^\eta}{\sum_{i, i \neq j} P(W_i, C_i, \lambda)^\eta}\right) \quad (5)$$

$$(\log P(c[1]|c[2..D]))^* = \log P(c[1]|c[2..D]) - \epsilon\gamma\ell(d_i)(1 - \ell(d_i)) \left(-I(C_j, c[1], c[2..D]) + \sum_{i, i \neq j} I(C_i, c[1], c[2..D]) \frac{P(W_i, C_i, \lambda)^\eta}{\sum_{i, i \neq j} P(W_i, C_i, \lambda)^\eta}\right) \quad (6)$$

$$(\log P(w|\mathbf{c}))^* = \log P(w|\mathbf{c}) - \epsilon\gamma\ell(d_i)(1 - \ell(d_i)) \left(-I(C_j, w, \mathbf{c}) + \sum_{i, i \neq j} I(C_i, w, \mathbf{c}) \frac{P(W_i, C_i, \lambda)^\eta}{\sum_{i, i \neq j} P(W_i, C_i, \lambda)^\eta}\right) \quad (7)$$

where $I(C_i, n, \mathbf{c}')$ denotes the number of times the operation of popping up n semantic tags at the current vector state $\mathbf{c}'$ in the C_i parse tree, $I(C_i, c[1], c[2..D])$ denotes the number of times the operation of pushing the semantic tag $c[1]$ at the current vector state $c[2..D]$ in the C_i parse tree and $I(C_i, w, \mathbf{c})$ denotes the number of times of emitting the word w at the state $\mathbf{c}$ in the parse tree C_i.

3 Experimental Setup and Results

GENIA (Kim et al., 2003) is a collection of 2000 research abstracts selected from the search results of MEDLINE database using keywords (MESH terms) *"human, blood cells and transcription factors"*. All these abstracts were then split into sentences and those containing more than two protein names and at least one interaction keyword were kept. Altogether 3533 sentences were left and 2500 sentences were sampled to build our data set.

The results using MLE and discriminative training are listed in Table 1. Discriminative training improves on the MLE by relatively 2.5% where N and I are set to 5 and 200 individually. Here N denotes the number of semantic parse hypotheses and I denotes the the number of sentences in the training data.

Table 1: Performance comparison of MLE versus Discriminative training

Measurement	GENIA	
	MLE	Discriminative
Recall	61.78%	64.59%
Precision	61.16%	61.51%
F-measure	61.47%	63.01%

References

J.K. Chen and F.K. Soong. 1994. An n-best candidates-based discriminative training for speech recognition applications. *IEEE Transactions on Speech and Audio Processing*, 2:206 – 216.

W. Chou, C.H. Lee, and B.H. Juang. 1993. Minimum error rate training based on n-best string models. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '93*, volume 2, pages 652 – 655.

Y. He and S. Young. 2005. Semantic processing using the hidden vector state model. *Computer Speech and Language*, 19(1):85–106.

B.H. Juang, W. Chou, and C.H. Lee. 1993. Statistical and discriminative methods for speech recognition. In Rubio, editor, *Speech Recognition and Understanding*, NATO ASI Series, Berlin. Springer-Verlag.

JD. Kim, T. Ohta, Y. Tateisi, and J Tsujii. 2003. GENIA corpus–semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(Suppl 1):i180–2.

D. Klein and C. D. Manning. 2002. Conditional structure versus conditional estimation in nlp models. In *Proc. the ACL-02 conference on Empirical methods in natural language processing*, pages 9–16, University of Pennsylvania, PA.

H.-K.J. Kuo, E. Fosle-Lussier, H. Jiang, and C.H. Lee. 2002. Discriminative training of language models for speech recognition. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '02*, volume 1, pages 325 – 328.

Deyu Zhou and Yulan He. 2008. Extracting Interactions between Proteins from the Literature. *Journal of Biomedical Informatics*, 41:393–407.

Deyu Zhou, Yulan He, and Chee Keong Kwoh. 2008. Extracting Protein-Protein Interactions from the Literature using the Hidden Vector State Model. *International Journal of Bioinformatics Research and Applications*, 4(1):64–80.

A preliminary approach to recognize generic drug names by combining UMLS resources and USAN naming conventions

Isabel Segura-Bedmar
Computer Sciences Department
Carlos III University of Madrid
Avd. Universidad, 30, Leganés,
28911, Madrid, Spain
isegura@inf.uc3m.es

Paloma Martínez
Computer Sciences Department
Carlos III University of Madrid
Avd. Universidad, 30, Leganés,
28911, Madrid, Spain
pmf@inf.uc3m.es

Doaa Samy
Linguistic Department
Cairo University
Egypt

dsamy@cu.edu.eg

Abstract

This paper presents a system[1] for drug name identification and classification in biomedical texts.

1 Introduction

Numerous studies have tackled gene and protein names recognition (Collier et al, 2002), (Tanabe and Wilbur, 2002). Nevertheless, drug names have not been widely addressed (Rindflesch et al., 2000).

Automating the process of new drugs recognition and classification is a challenging task. With the rapidly changing vocabulary, new drugs are introduced while old ones are made obsolete. Though the terminological resources are frequently updated, they can not follow the accelerated pace of the changing terminology.

Drug receives three distinct names: the chemical name, the generic (or nonproprietary) name, and the brand (or trademark) name. The U.S. Adopted Name (USAN) Council establishes specific nomenclature rules for naming generic drugs. These rules rely on the use of affixes that classify drugs according to their chemical structure, indication or mechanism of action. For example, analgesics substances can receive affixes such as *-adol-, -butazone, -fenine, -eridine* and *–fentanil*. In the present work, we focus, particulary, on the implementation of a set of 531 affixes approved by the USAN Council and published in 2007[2]. The affixes allow a specific classification of drugs on pharmacological families, which ULMS Semantic NetWork is unable to provide.

2 The System

The system consists of four main modules: a basic text processing module, WordNet look-up module, UMLS look-up module and the USAN rules module, as shown in Figure 1.

A corpus of 90 medical abstracts was compiled for the experiment. For the basic processing of the abstracts, GATE[3] architecture is used. This text processing provides sentence segmentation, tokenization and POS tagging. Tokens which receive a noun or proper noun POS tag are extracted.

The nouns found on WordNet are discarded and those which are not found in WordNet are looked up in the UMLS Metathesaurus. If a noun is found in UMLS, it is tagged with its corresponding semantic types as assigned by UMLS. A subset of these nouns is tagged as "drug" if their semantic types are "Pharmacological Substance" or "Antibiotic". Finally, nouns which have not been found in UMLS are tagged as "unknown".

The list of nouns tagged as "drug" is passed to the rule module to detect their pharmacological families according to the affixes. In addition, the rule module processes the list of "unknown" nouns which are not found in UMLS to check the presence of affixes, and thereby, of possible drugs.

3 Preliminary results

[1] This work has been partially supported by the projects: FIT-350300-2007-75 (Semantic Interoperability in Electronic Health Care) and TIN2007-67407-C03-01 (BRAVO: Advanced Multimodal and Multilingual Question Answering).

[2] http://www.ama-assn.org/ama1/pub/upload/mm/365/usan_stem_list.pdf Accessed January 2008

[3] http://www.gate.ac.uk/

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 100–101,
Columbus, Ohio, USA, June 2008.

A manual evaluation by a domain[4] expert was carried out. The list of nouns not found in WordNet contained 1885 initial candidates. This initial list is looked up in UMLS and 93.4% of them (1761) is linked with some concepts of UMLS. The UMLS module recognized 1400 nouns as pharmacological substances or antibiotics. The rest of nouns, 361, are detected by UMLS but neither as pharmacological substance nor as antibiotics.

The expert manually evaluated the set of nouns detected by UMLS as pharmacological substances or antibiotics (1400). Evaluation showed that only 1100 were valid drugs.

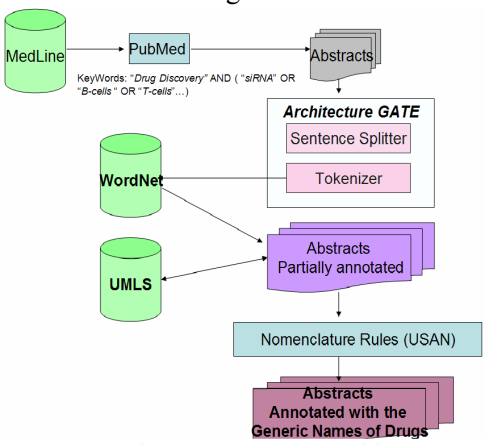

Figure 1 System Architecture

The list of nouns (124) which have not been found in UMLS are processed by the rule module to detect new candidate drugs not included in UMLS. This module only detects 17 candidate drugs. The manual evaluation showed that 7 of them were valid drugs and the rest of nouns are biomedical concepts not included in UMLS. Some of these drugs are *Mideplanin*, *Tomopenem*, *Elvitegravir*, and so on. The rest of nouns neither detected by the UMLS module nor by the rules module, 106, were also validated by the expert in order to estimate the overall coverage of our approach. The evaluation of these nouns shows that only 7 of them are valid drugs, however, the rest of the nouns are named entities of the general domain (organization, person names or cities) or biomedical concepts. Introducing a module of generic NER should decrease the noise caused by such entities.

Finally, precision and recall of the overall system combining UMLS and rules were calculated. The system achieved 78% of precision and 99.3% of recall

3.1 The classification in pharmacological families

Once processed by the rule module, 73.8% of the candidate drugs recognised by UMLS were also classified in pharmacological families by the USAN naming rules. Expert's evaluation of the rule-based classification showed that rules achieved 89% precision. Short affixes such as –ol, –pin and -ox are responsible of the wrong classifications. Thus, additional clues are necessary to detect these drug families.

4 Some Conclusions

As a preliminary approach, it is a first step towards a useful Information Extraction System in the field of Pharmacology. Though evaluation reveals that rules alone are not feasible enough in detecting drugs, but they help to improve the coverage. In addition, rules provide a drug classification in pharmacological families. Such classification is an added value in the development of NLP applications within the pharmacological domain.

For future work, the approach will be extended to address additional information about pharmacologic classes included in many biomedical terminologies integrated in the UMLS such as MeSH or SNOMED.

Future work will also target a wider coverage and a bigger set of drug types through including more affixes, detecting complex entities (multi-words), detecting synonyms, resolving acronyms and ambiguities as well as using contextual information to disambiguate the correct semantic type of each term occurring in the texts.

References

Collier N, Takeuchi K. 2004. Comparison of characterlevel and part of speech features for name recognition in bio-medical texts:423− 35.

Rindflesch, T.C., Tanabe,L., Weinstein,J.N. and Hunter,L. 2000. EDGAR: extraction of drugs, genes and relations from the biomedical literature. Pac. Symp. Biocomput. 5, 517–528

Tanabe, L. y Wilbur, W.J. 2002. Tagging gene and protein names in biomedical text. Bioinformatics 18, 1124–1132

[4] The authors are grateful to Maria Bedmar Segura, Manager of the Drug Information Center, Mostoles University Hospital, for her valuable assistance in the evaluation of the system.

Mapping Clinical Notes to Medical Terminology at Point of Care

Yefeng Wang
School of Information Technologies
University of Sydney
New South Wales 2006, Australia
ywang1@it.usyd.edu.au

Jon Patrick
School of Information Technologies
University of Sydney
New South Wales 2006, Australia
jonpat@it.usyd.edu.au

Abstract

Clinicians write the reports in natural language which contains a large amount of informal medical term. Automating conversion of text into clinical terminologies allows reliable retrieval and analysis of the clinical notes. We have created an algorithm that maps medical expressions in clinical notes into a medical terminology. This algorithm indexes medical terms into an augmented lexicon. It performs lexical searches in text and finds the longest possible matches in the target terminology, SNOMED CT. The mapping system was run on a collection of 470,000 clinical notes from an Intensive Care Service (ICS). The evaluation on a small part of the corpus shows the precision is 70.4%.

1 Introduction

A substantial amount of clinical data is locked away in a non-standardised form of clinical language which if standardised could be usefully mined to gain greater understanding of patient care and the progression of diseases. Clinical notes on a patient's health are written in natural language which contains a great deal of formal terminology but used in an informal and unorderly manner. These medical notes need to be converted to a formal terminology to enable accurate retrieval and to compile aggregated statistics of the medical care. To satisfy these needs, we developed a medical concept identifier that is able to identify concepts in clinical notes and mapped to medical codes in a terminology. The algorithm has been implemented to tag medical concepts in a collection of 470,000 clinical notes from an Intensive Care Service. A total of 9,135,000 instances of about 20,000 medical concepts were identified. These medical concepts are used to study the medical language used by Intensive Care clinical staff, and the identified concepts are used to index patient clinical records for targeted information retrieval activities.

2 Related Work

There has been a large effort spent on automatic recognition of medical and biomedical concepts and mapping them to medical terminology. The Unified Medical Language System Meta-thesaurus (UMLS) is the world's largest medical knowledge source and it has been the focus of much research. One of the prominent systems to map free text to UMLS are MetaMap (Aronson, 2001),

3 Constructing the Lexicon

The Augmented Lexicon is a data structure developed to keep track of the words that appear in the concepts of the medical terminology. The Augmented Lexicon is built from the individual words in the gloss or the definition of the medical term. For example, *Myocardial Infarction* has the atomic words *Myocardial* and *Infarction*. Each concept is normalised which includes removal of stop words, stemming, and spelling variation generation. For each word, a list of the concept ids that contain that word is stored in the Augmented Lexicon. An additional table is stored alongside the augmented lexicon, called the "atomic term count" to record the number of atomic terms that comprise each description.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 102–103,
Columbus, Ohio, USA, June 2008.

4 Token Matching Algorithm

The algorithm performs string alignment between the source text and a target medical terminology. The best matches are determined by scoring algorithms for both perfect matching and partial matching. To find all possible matches, the algorithm iteratively performs matches for sub-strings using dynamic programming, so that the algorithm doesn't have to generate all combination of sub-strings for the input sentence. Each previously computed substrings matches are stored and in a matching matrix so don't require recalculation.

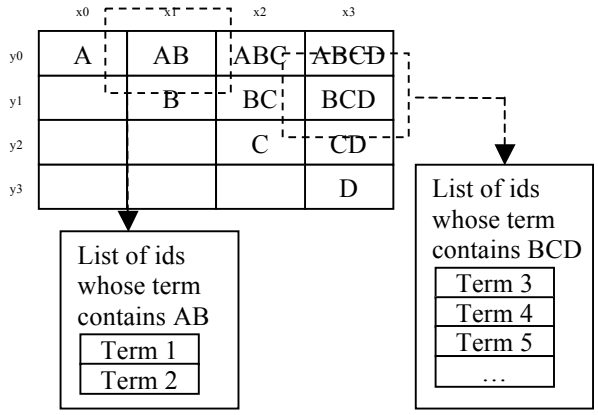

Figure 1: Token Matching Matrix

The data stored in each cell is a list of medical term ids that are in all the tokens that comprise the cell. The score is then calculated using the "atomic term count", which stores the number of tokens that make up that term. The score is the number of tokens in the current cell that have the term id in common divided by the number of tokens in the full description.

5 Recognition of Clinical Entities

Before medical term identification, Clinical entities such as measurement, demography, quantities are recognised and normalised to their classes.

Entity Class	Examples
Blood Pressure	105mm of Hg
Demography	69 year-old man
Datetime	20/11 2030
Quantity	55 mm

Table 1. Clinical Entities and Examples

6 Evaluation

The token matching algorithm has been implemented as a module in a terminology server that can provide real time text to medical concept encoding. The system was installed in the Intensive Care Service that provides web interfaces for users to submit clinical notes and it computed SNOMED CT codes in real-time. The web interface has been implemented in several clinical forms templates at the RPAH, allowing data to be captured as the doctors fill in these forms. A feedback form has been implemented allowing clinicians to submit comments, identify terms that are missed by the system and submit corrections to incorrectly labeled terms. This was seen as a rare opportunity to collect an expert corrected corpus of clinical notes. Unfortunately, there was little adherence to the correction part of the program and so we do not yet have sufficient material to be precise about recall values.

To evaluate the accuracy our systems, we collected a set of bedside clinical notes of patient monitoring chart information. 487 documents and 4,054 medical concepts were tagged with SNOMED CT codes and have been evaluated by medical experts. There are 2,852 correctly identified concepts and 1,202 incorrectly identified concepts, results in a precision rate of 70.4%. The recall rate hasn't been fully evaluated.

7 Conclusions

In conclusion, we have proposed a system to find medical terms in free text clinical notes and map them into a medical terminology. We have implemented the algorithm as a web-service system. The algorithm uses an augmented lexicon to index concept descriptors in SNOMED CT, which allow a much faster mapping of longest spanning concepts in the system than a naïve word searching approach, which can then create more effective information retrieval and information extraction.

References

Aronson, A. R. (2001). *Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program*. Proc AMIA Symp 17: 21.

An Approach to Reducing Annotation Costs for BioNLP

Michael Bloodgood
Computer and Information Sciences
University of Delaware
Newark, DE 19716
bloodgoo@cis.udel.edu

K. Vijay-Shanker
Computer and Information Sciences
University of Delaware
Newark, DE 19716
vijay@cis.udel.edu

1 Introduction

There is a broad range of BioNLP tasks for which active learning (AL) can significantly reduce annotation costs and a specific AL algorithm we have developed is particularly effective in reducing annotation costs for these tasks. We have previously developed an AL algorithm called *ClosestInitPA* that works best with tasks that have the following characteristics: redundancy in training material, burdensome annotation costs, Support Vector Machines (SVMs) work well for the task, and imbalanced datasets (i.e. when set up as a binary classification problem, one class is substantially rarer than the other). Many BioNLP tasks have these characteristics and thus our AL algorithm is a natural approach to apply to BioNLP tasks.

2 Active Learning Algorithm

ClosestInitPA uses SVMs as its base learner. This fits well with many BioNLP tasks where SVMs deliver high performance (Giuliano et al., 2006; Lee et al., 2004). *ClosestInitPA* is based on the strategy of selecting the points which are closest to the current model's hyperplane (Tong and Koller, 2002) for human annotation. *ClosestInitPA* works best in situations with imbalanced data, which is often the case for BioNLP tasks. For example, in the AIMed dataset annotated with protein-protein interactions, the percentage of pairs of proteins in the same sentence that are annotated as interacting is only 17.6%.

SVMs (Vapnik, 1998) are learning systems that learn linear functions for classification. A statement of the optimization problem solved by soft-margin SVMs that enables the use of asymmetric cost factors is the following:

Minimize: $\frac{1}{2}\| \vec{w} \|^2 + C_+ \sum_{i:y_i=+1} \xi_i + C_- \sum_{j:y_j=-1} \xi_j$ (1)

Subject to: $\forall k : y_k[\vec{w} \cdot \vec{x}_k + b] \geq 1 - \xi_k$ (2)

where $(\vec{w}, b)$ represents the hyperplane that is learned, $\vec{x}_k$ is the feature vector for example k, y_k in {+1,-1} is the label for example k, $\xi_k = \max(0, 1 - y_k[\vec{w} \cdot \vec{x}_k + b])$ is the slack variable for example k, and C_+ and C_- are user-defined cost factors that trade off separating the data with a large margin and misclassifying training examples.

Let PA=C_+/C_-. PA stands for "positive amplification." We use this term because as the PA is increased, the importance of positive examples is amplified. *ClosestInitPA* is described in Figure 3. We have previously shown that setting PA based on a small initial set of data outperforms the more obvious approach of using the current labeled data to estimate PA.

Initialization:
- *L* = small initial set of labeled data
- *U* = large pool of unlabeled data

$$PA = \frac{\#\text{ neg examples in } L}{\#\text{ pos examples in } L}$$

Loop until stopping criterion is met:
1. Train an SVM with parameters C_+ and C_- set such that C_+/C_- = PA.
2. *batch* = select k points from *U* that are closest to the hyperplane learned in step 1.
   U = *U* – *batch*
   L = *L* U *batch*

Figure 3. *ClosestInitPA* algorithm.

We have previously developed a stopping criterion called *staticPredictions* that is based on stopping when we detect that the predictions of our models on some unlabeled data have stabilized. All of the automatic stopping points in our results are determined using *staticPredictions*.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 104–105,
Columbus, Ohio, USA, June 2008.

3 Experiments

Protein-Protein Interaction Extraction: We used the AImed corpus, which was previously used for training protein interaction extraction systems in (Giuliano et al., 2006). We cast RE as a binary classification task as in (Giuliano et al., 2006).

We do 10-fold cross validation and use what is referred to in (Giuliano et al., 2006) as the K_{GC} kernel with SVM^{light} (Joachims, 1999) in our experiments. Table 1 reports the results.

StoppingPoint	Average # Labels	F Measure	
		Random	AL
20%	1012	48.33	54.34
30%	1516	49.76	54.52
40%	2022	53.11	56.39
100%	5060	57.54	57.54
AutoStopPoint	1562	51.25	55.34

Table 1. AImed Stopping Point Performance. "AutoStopPoint" is when the stopping criterion says to stop.

Medline Text Classification: We use the Ohsumed corpus (Hersh, 1994) and a linear kernel with SVM^{light} with binary features for each word that occurs in the training data at least three times. Results for the five largest categories for one versus the rest classification are in Table 2.

StoppingPoint	Average # Labels	F Measure	
		Random	AL
20%	1260	49.99	61.49
30%	1880	54.18	62.72
40%	2500	57.46	63.75
100%	6260	65.75	65.75
AutoStopPoint	1204	47.06	60.73

Table 2. Ohsumed stopping point performance. "AutoStopPoint" is when the stopping criterion says to stop.

GENIA NER: We assume a two-phase model (Lee et al., 2004) where boundary identification of named entities is performed in the first phase and the entities are classified in the second phase. As in the semantic classification evaluation of (Lee et al., 2004), we assume that boundary identification has been performed. We use features based on those from (Lee et al., 2004), a one versus the rest setup and 10-fold cross validation. Tables 3-5 show the results for the three most common types in GENIA.

StoppingPoint	Average # Labels	F Measure	
		Random	AL
20%	13440	86.78	90.16
30%	20120	87.81	90.27
40%	26900	88.55	90.32
100%	67220	90.28	90.28
AutoStopPoint	8720	85.41	89.24

Table 3. Protein stopping points performance. "AutoStopPoint" is when the stopping criterion says to stop.

StoppingPoint	Average # Labels	F Measure	
		Random	AL
20%	13440	79.85	82.06
30%	20120	80.40	81.98
40%	26900	80.85	81.84
100%	67220	81.68	81.68
AutoStopPoint	7060	78.35	82.29

Table 4. DNA stopping points performance. "AutoStopPoint" is when the stopping criterion says to stop.

StoppingPoint	Average # Labels	F Measure	
		Random	AL
20%	13440	84.01	86.76
30%	20120	84.62	86.63
40%	26900	85.25	86.45
100%	67220	86.08	86.08
AutoStopPoint	4200	81.32	86.31

Table 5. Cell Type stopping points performance. "AutoStopPoint" is when the stopping criterion says to stop.

4 Conclusions

ClosestInitPA is well suited to many BioNLP tasks. In experiments, the annotation savings are practically significant for extracting protein-protein interactions, classifying Medline text, and performing biomedical named entity recognition.

References

Claudio Giuliano, Alberto Lavelli, and Lorenza Romano. 2006. Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. In *Proceedings of the EACL*, 401-408.

William Hersh, Buckley, C., Leone, T.J., and Hickman, D. (1994). Ohsumed: an interactive retrieval evaluation and new large text collection for research. ACM SIGIR.

Thorsten Joachims. 1999. Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*, MIT-Press, 169-184.

Ki-Joong Lee, Young-Sook Hwang, Seonho Kim, and Hae-Chang Rim. 2004. Biomedical named entity recognition using two-phase model based on SVMs. *Journal of Biomedical Informatics, Vol 37*, 436–447.

Simon Tong and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *JMLR* 2: 45-66.

Vladimir Vapnik. 1998. *Statistical Learning Theory*. John Wiley & Sons, New York, NY, USA.

Temporal Annotation of Clinical Text

Danielle L. Mowery MS, Henk Harkema PhD, Wendy W. Chapman PhD
Department of Biomedical Informatics
University of Pittsburgh, Pittsburgh, PA 15260, USA
dlm31@pitt.edu, heh23@pitt.edu, wec6@pitt.edu

Abstract

We developed a temporal annotation schema that provides a structured method to capture contextual and temporal features of clinical conditions found in clinical reports. In this poster we describe the elements of the annotation schema and provide results of an initial annotation study on a document set comprising six different types of clinical reports.

1 Introduction

Distinguishing between historical and recent conditions is important for most tasks involving retrieval of patients or extraction of information from textual clinical records. Various approaches can be used to determine whether a condition is historical or recent. Chapman et al. (2007) developed an algorithm called ConText that uses trigger terms like "history" to predict whether a condition is historical. Studies of ConText show that this approach is inadequate for determining whether a condition is historical, achieving recall of 67% and precision 74% on emergency department reports. Temporal modeling methods commonly reason about the temporality of an event with respect to absolute time and other temporally related events (Zhou et al., 2006; Chambers et al., 2007). Knowing the relative or absolute time the condition occurred can be useful in determining whether the condition is historical. However, we hypothesize that many clinical conditions in clinical reports are not modified by explicit temporal references.

To test this hypothesis and explore other types of information that may be useful in automatically distinguishing historical from recent clinical conditions in dictated clinical records, we developed a temporal annotation schema that accounts for explicit temporal expressions, temporal trigger terms, and clinical reporting acts described in reports. Three annotators applied the schema to six types of reports. We measured inter-annotator agreement scores and obtained prevalence and distribution figures for the three annotation types.

2 Methods

2.1 Dataset

Our dataset is comprised of 24 clinical reports of six types dictated at the University of Pittsburgh Medical Center during 2007: discharge summaries, surgical pathology, radiology, echocardiograms, operative gastrointestinal, and emergency department reports. A physician pre-annotated the 518 clinical conditions in the reports and marked each one as recent or historical.

We developed our annotation schema using one of each report type (six reports). Annotators (authors HH, DM and WC) annotated the remaining 18 reports as described below.

2.2 Annotation Schema

For our temporal annotation study, each pre-annotated clinical condition was annotated with three types of information: temporal expression, trigger term, and clinical reporting act.

The set of **temporal expressions** (TEs) is taken from Zhou et al. (2006) and includes categories such as DATE AND TIME for explicit TEs and KEY EVENTS for TEs relative to significant clinical events. A given clinical condition is annotated with the category of the TE it is modified by. For example, in the sentence *"The stroke occurred on 1/5/2000"*, the condition "stroke" is annotated with category DATE AND TIME. There is also a category NO TEMPORAL EXPRESSION for annotating conditions that are not linked to a TE.

Trigger terms (TTs) are explicit signals (words and phrases) in text other than TEs that indicate

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 106–107,
Columbus, Ohio, USA, June 2008.

whether a condition is recent or historical (Chapman et al., 2007). If a condition co-occurs with a TT, it is annotated with TRIGGER: YES. For example, "pneumonia" in the sentence *"Films indicate pneumonia, which is new for this patient"* is annotated as TRIGGER: YES because "new" is a TT.

Error analyses of our previous studies indicate that the context in which a condition is mentioned in a report is potentially useful for prediction of a condition as recent or historical. Clinical reports consist of statements that group into segments according to the **clinical reporting act** (CRA) they describe, such as noting a past history and considering a diagnosis. CRAs are tightly correlated with report sections; however, sections are not consistent, and different CRAs can occur within a single section. We distinguish 16 CRAs. Each clinical condition is annotated with one CRA. For example, the condition "smoker" in the sentence *"She was a smoker"* is annotated SOCIAL HISTORY.

2.3 Analysis

To establish the level of inter-annotator agreement, we iteratively annotated groups of six reports (one of each type). After each iteration, we refined our annotation schema and guidelines. We analyzed annotations, overall and by report type, in the following way: 1) calculate inter-annotator kappa score, 2) measure prevalence of TT and TE categories, and 3) observe distribution of CRAs.

3 Results and Discussion

As shown in figure 1, average inter-annotator scores as measured by Cohen's kappa for TE, TT, and CRA (.68, .82 and .72 respectively) reached acceptable levels after three iterations and are expected to rise further with increased annotation experience and understanding of the guidelines.

Table 1 shows the prevalence of TEs and TTs across six report types, where prevalence is defined as the frequency of TE or TT found in a given report. Use of TEs across report types ranged from 0% to 52% whereas TTs were found less often at 0% to 34% by report genre. Table 2 plots the correlation between the CRA assigned to a clinical condition and the condition's classification as recent or historical. We found that there is a strong correlation for the most commonly occurring clinical reporting acts (PH, PR, and PO). We are therefore optimistic that CRAs can serve as an informative feature for a statistical recent/historical classifier.

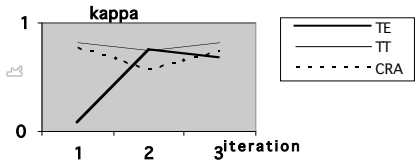

Figure 1. Average Cohen's kappa agreement for 3 iterations.

	DS	E	ED	GI	RAD	SP	O
TE	48(52)	0(0)	51(20)	2(10)	1(5)	8(36)	110(21)
TT	32(34)	0(0)	54(21)	1(5)	0(0)	6(27)	93(17)

Table 1. Prevalence, count (%), of TE and TT across report types, overall. DS: *discharge summary,* E: e*chocardiogram,* ED: *emergency department,* GI: *operative gastrointestinal*, RAD: *radiology*, SP: *surgical pathology* and O: *overall.*

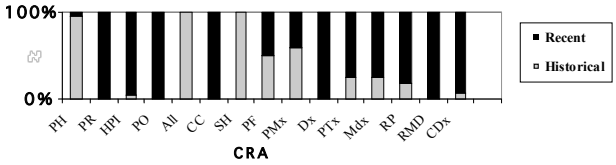

Table 2. Historical/recent distribution of CRAs. PH: *Past history,* PR*, Patient reporting,* HPI: *History of present illness,* PO: *Physician observing,* All: *Allergies,* CC: *Chief complaint,* SH: *Social history,* FH: *Family history,* PF: *Past Finding*, PMx, *Past medication,* Dx: *Diagnosis,* PTx: *Plan treatment,* Mdx: *Prescribing medication*, RP: *Referring problem,* RMD: *Refer to MD,* CDx: *Considering diagnosis.*

The finding that many conditions are associated with neither a TE nor a TT and study of ConText's limitations with such categories at the scope of the sentence suggests that additional features are necessary to discern a condition as recent or historical. Whereas temporality in discourse may follow a sequential chronology as narrative unfolds, references to past instances within clinical text are not easily resolved. We are optimistic that CRAs may help this issue and will focus our study to evaluate whether these three features are sufficient together.

References

L. Zhou, G. B. Melton, S. Parsons, G. Hripcsak. 2006. A temporal constraint structure for extracting temporal information from clinical narrative. *Journal of Biomedical Informatics*, 39(4):424-439

N. Chambers, S. Wang, D. Jurafsky. 2007. Classifying Temporal Relations Between Events. In: *ACL-07*.

W. Chapman, D. Chu, J. N. Dowling. 2007. ConText: An Algorithm for Identifying Contextual Features from Clinical Text. In: *ACL-07*.

CBR-Tagger: a case-based reasoning approach to the gene/protein mention problem

Mariana Neves
Biocomputing Unit
Centro Nacional de Biotecnología - CSIC
Madrid, 28049, Spain
mlara@cnb.csic.es

Monica Chagoyen
Biocomputing Unit
Centro Nacional de Biotecnología - CSIC
Madrid, 28049, Spain
monica.chagoyen@cnb.csic.es

José M. Carazo
Biocomputing Unit
Centro Nacional de Biotecnología - CSIC
Madrid, 28049, Spain
carazo@cnb.csic.es

Alberto Pascual-Montano
Departamento de Arquitectura de Computadores
Facultad de Ciencias Físicas, UCM
Madrid, 28040, Spain
pascual@fis.ucm.es

Abstract

This work proposes a case-based classifier to tackle the gene/protein mention problem in biomedical literature. The so called gene mention problem consists of the recognition of gene and protein entities in scientific texts. A classification process aiming at deciding if a term is a gene mention or not is carried out for each word in the text. It is based on the selection of the best or most similar case in a base of known and unknown cases. The approach was evaluated on several datasets for different organisms and results show the suitability of this approach for the gene mention problem.

1 Introduction

This paper proposes a new method to the gene mention problem by using a case-based reasoning approach that performs a binary classification (gene mention or not) for each word in a text. In a first step cases are stored in two bases (known and unknown cases), followed by a search in these bases for the case most similar to the problem. The classification decision is given by the class of the case selected. The system was developed using Java and MySQL technologies and is available for download as part of the Moara project[1].

2 Proposed method

The method here proposed identifies gene mentions in a text by means of classifying each token into two possible classes: gene mention or not. The system consists of two main steps: the construction of the case bases, and the testing phase, when the test dataset is presented to the system to identify the possible mentions. The words extracted from the training documents were the tokens used to construct the two case bases, one for known cases and the other for unknown cases, as proposed for the part-of-speech tagging problem in (Daelemans, Zavrel, Berck, & Gillis, 1996).

The known cases are the ones used by the system to classify those words that are not new, i.e. those that have were present in the training dataset. The attributes used to represent a known case are the word itself, the class of the word (if it is a gene mention or not), and the class of the preceding word (if it is a gene mention or not).

The system uses a second case base to decide about words that are unknown to the system, i.e. those that are not present in the training set. The attributes of the unknown cases were the shape of the word, the class of the word (if it is a gene mention or not), and the class of the preceding word (if it is a gene mention or not). Note that instead of saving the word itself, a shape of the word is kept in order to allow the system to be able to classify unknown words by means of looking for cases with similar shape. The shape of the word is given by its transformation in a set of symbols according to the type of character found.

In the construction of cases, each word represents a single case, and in order to account for repetitions, the frequency of the case is incremented to indicate the number of times that it appears in the training dataset. The training

[1] http://biocomp.cnb.csic.es/~mlara/moara/index.html

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 108–109,
Columbus, Ohio, USA, June 2008.

documents are read twice, one in the forward (from left to right), and one in the backward (from right to left) directions, in order to allow a more variety of cases. This is important as the classification of a token may be influenced by its preceding and following words.

CBR-Tagger has also been trained with additional corpora in order to better extract mentions from different organisms. These extra corpora are the datasets for gene normalization of the BioCreative task 1B (Hirschman, Colosimo, Morgan, & Yeh, 2005) for to yeast, mouse and fly and the BioCreative 2 Gene Normalization task (Morgan & Hirschman, 2007) for human.

In the classification procedure, the text is tokenized and a sliding window is applied first in the forward and then in the backward direction. In each case, the system keeps track of the class of the preceding token (false at the beginning), gets the shape of the token and tries to find in the bases a case most similar to it. The search procedure is divided in two parts, for the known and unknown cases. Priority is always given to the known cases since it saves the word exactly as they appeared in the training documents and the classification may be more precise than using the unknown cases.

A token already classified as positive by the forward reading may be used for the backward reading as preceding class and might help recognizing mentions composed by many tokens that would not have been totally recognized by one of the reading procedures only. After the identification of the best case for each token, some post-processing procedures are executed to check boundaries (for mentions composed of more than one token) as well as abbreviations and corresponding full names.

3 Results

The results obtained with the BioCreative 2 gene mention task for the CBR-Tagger are shown in Table 1 along with the best result of the competition. Results are showed according to the datasets used for the training of the CBR-tagger: BioCreative 2 Gene Mention task (Wilbur, Smith, & Tanabe, 2007) corpus only (CbrBC2), and the combination of it with the BioCreative task 1B gene normalization corpus (Hirschman et al., 2005) for the yeast (CbrBC2y), mouse (CbrBC2m), fly (CbrBC2f) and the three of them (CbrBC2ymf).

Taggers	*P*	*R*	*FM*
CbrBC2	77.8	75.9	76.9
CbrBC2y	82.7	52.6	64.7
CbrBC2m	83.1	47.1	60.1
CbrBC2f	82.0	65.9	73.0
CbrBC2ymf	82.5	39.7	53.6
Best BC2 result	88.5	86.0	87.2

Table 1: Results for the BC2 gene mention task.

CBR-Tagger has also been applied to the gene normalization problem in conjunction with two other available taggers: Abner[2] and Banner[3]. Table 2 summarizes the best mix of taggers configuration for each organism. Detailed results may be found at the author's research page[4].

Organism	*Best configuration*
Yeast	Abner+CbrBC2
Mouse	Abner+CbrBC2m
Fly	CbrBC2f
Human	Banner+CbrBC2ymf

Table 2: Best taggers for each organism.

Acknowledgments

This work has been partially funded by the Spanish grants BIO2007-67150-C03-02, S-Gen-0166/2006, TIN2005-5619. APM acknowledges the support of the Spanish Ramón y Cajal program. The authors acknowledge support from Integromics, S.L.

References

Daelemans, W., Zavrel, J., Berck, P., & Gillis, S. (1996). *MBT: A Memory-Based Part of Speech Tagger-Generator.* Paper presented at the Fourth Workshop on Very Large Corpora, Copenhagen, Denmark.

Hirschman, L., Colosimo, M., Morgan, A., & Yeh, A. (2005). Overview of BioCreAtIvE task 1B: normalized gene lists. *BMC Bioinformatics, 6 Suppl 1*, S11.

Morgan, A., & Hirschman, L. (2007). *Overview of BioCreative II Gene Normalization.* Paper presented at the Second BioCreative Challenge Evaluation Workshop, Madrid-Spain.

Wilbur, J., Smith, L., & Tanabe, L. (2007). *BioCreative 2. Gene Mention Task.* Paper presented at the Second BioCreative Challenge Evaluation Workshop, Madrid, Spain.

[2] http://pages.cs.wisc.edu/~bsettles/abner/
[3] http://banner.sourceforge.net/
[4] http://biocomp.cnb.csic.es/~mlara/mention.html

Textual Information for Predicting Functional Properties of the Genes

Oana Frunza and Diana Inkpen
School of Information Technology and Engineering
University of Ottawa Ottawa, ON, Canada, K1N 6N5
{ofrunza,diana}@site.uottawa.ca

1 Overview

This paper is focused on determining which proteins affect the activity of *Aryl Hydrocarbon Receptor (AHR*) system when learning a model that can accurately predict its activity when single genes are knocked out. Experiments with results are presented when models are trained on a single source of information: abstracts from Medline (http://medline.cos.com/) that talk about the genes involved in the experiments. The results suggest that AdaBoost classifier with a binary bag-of-words representation obtains significantly better results.

2 Task Description and Data Sets

The task that we address is a biology-specific task considered a competition track for KDDCup2002 (http://www.biostat.wisc.edu/~craven/kddcup/winners.html).

The organizers of the KDD Cup competition provided data obtained from experiments performed on a set of yeast strains in which each strain contains a single gene that is knocked out (a gene sequence in which a single gene is inoperative). Each experiment had associated a discretized value of the activity of the AHR system when a single gene was knocked out. 3 possible classes describe the systems' response. The "**nc**" label indicates that the activity of the hidden system was not significantly different than the baseline (the wild-type yeast); the "**control**" label indicates that the activity was significantly different than the baseline for the given instance, and that the activity of another hidden system (the control) was also significantly changed compared to its baseline; the "**change**" label shows that the activity of the hidden system was significantly changed, but the activity of the control system was not significantly changed.

The organizers of the KDD Cup evaluate the task as a two-class problem with focus on the positive class. The first definition is called the "**narrow**" definition of the positive class and it is specific to the knocked-out genes that had an AHR-specific effect. In this case the positive class is defined by the experiments in which the label of the system is "*change"* and the negative examples are the experiments that consist of those genes with either the "*nc*" or the "*control*" label. The second definition consists of those genes labeled with either the "*change*" or the "*control*" label. The negative class consists of those genes labeled with the "*nc*" label. The second partitioning corresponds to the "**broad"** characterization of the positive class genes that affect the hidden system.

The area under the Receiver Operating Characteristic (ROC) - AUC curve is chosen as an evaluation measure. The global score for the task will be the summed AUC values for both the "narrow" and the "broad" partition of the data.

The sources of information provided by the organizers of the task contain: hierarchical information about the function and localization of the genes; relational information describing the protein-protein interactions; and textual information in abstracts from Medline that talk about the genes. Some characteristics of the data need to be taken into consideration in order to make suitable decisions for choosing the trainable system/classifier, the representation of the data, etc. Missing information is a characteristic of the data set. Not all genes had the location and function annotation, the protein-protein interaction information, or abstracts associated with the gene name. Besides the missing information, the high class imbalance is another fact that needs to be taken into account.

From the data that was released for the KDD competition we run experiments only with the genes that had associated abstracts. Table 1 presents a summary of the data sets used in our experiments after considering only the genes that had abstracts associated with them. The majority of the genes had one abstract, while others had as many as 22 abstracts.

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 110–111,
Columbus, Ohio, USA, June 2008.

Table 1. Summary of the data for our experiments with the two definitions of the positive class. In brackets are the original sizes of the data sets.

Data set	Narrow		Broad	
	Pos	Neg	Pos	Neg
Training	24 (37)	1,435 (2,980)	51 (83)	1,408 (2,934)
Test	11 (19)	715 (1,469)	30 (43)	696 (1,445)

3 Related Work

Previous research on the task was done by the teams that participated in the KDD Cup 2002. The textual information available in the task was considered as an auxiliary source of information and not the primary one, as in this article.

The winners of the task, Kowalczyk and Raskutti (2002) used the textual information as additional features to the ones extracted from other available information for the genes. They used a "bag-of-words" representation, removed stop words and words with low frequency. They used Support Vector machine (SVM) as a classifier.

Krogel *et. al*. (2002) used the textual information with an information extraction system in order to extract missing information (function, localization, protein class) for the genes in the released data set.

Vogel and Axelrod (2002) used the Medline abstracts to extract predictive keywords, and added them to their global system.

Our study investigates and suggests a textual representation and a trainable model suitable for this task and similar tasks in the biomedical domain.

4 Method

The method that we propose to solve the biology task is using Machine Learning (ML) classifiers suitable for a text classification task and various feature representations that are known to work well for data sets with high class imbalance. The task becomes a two-class classification: "**Positive**" versus "**Negative**", with a "**narrow**" and "**broad**" definition for the positive class. As classification algorithms we used: Complement Naive Bayes (CNB), AdaBoost, and SVM all from the Weka toolkit (http://www.cs.waikato.ac.nz/ml/weka/). Similar to the evaluation done for the KDD Cup, we consider the sum of the 2 AUC measures for the definitions of the positive class as an evaluation score. The random classifier with an AUC measure of 0.5 is considered as a baseline.

As a representation technique we used binary and frequency values for features that are: words extracted from the abstracts (bag-of-words (BOW) representation), UMLS concepts and UMLS phrases identified using the MetaMap system (http://mmtx.nlm.nih.gov/), and UMLS relations extracted from the UMLS metathesaurus. We also ran experiments with feature selection techniques.

Table 2 presents our best results using AdaBoost classifier for BOW, UMLS concepts, and UMLS relations representation techniques. "B" stands for binary and "Freq" stands for frequency counts.

Table 2. Sum of the AUC results for the two classes without feature selection.

Representation	**AdaBoost (AUC) Narrow**	**AdaBoost (AUC) Broad**	**Sumed AUC**
BOW_B	**0.613**	**0.598**	**1.211**
BOW_Freq	0.592	0.557	1.149
UMLS_B	0.571	0.607	1.178
UMLS_Freq	0.5	0.606	1.106
UMLS_Rel_B	0.505	0.547	1.052
UMLS_Rel_Freq	0.5	0.5	1

5 Discussion and Conclusion

Looking at the obtained results, a general conclusion can be made: textual information is useful for biology-specific tasks. Not only that it can improve the results but can also be considered a stand-alone source of knowledge in this domain. Without any additional knowledge, our result of 1.21 AUC sum is comparable with the sum of 1.23 AUC obtained by the winners of the KDD competition.

References

Adam Kowalczyk and Bhavani Raskutti, 2002. *One Class SVM for Yeast Regulation Prediction,* ACM SIGKDD Explorations Newsletter, Volume 4, Issue 2, pp. 99-100.

Mark A Krogel, Marcus Denecke, Marco Landwehr, and Tobias Scheffer. 2002. *Combining data and text mining techniques for yeast gene regulation prediction: a case study,* ACM SIGKDD Explorations Newsletter, Volume 4, Issue 2, pp. 104-105.

David S. Vogel and Randy C. Axelrod. 2002. *Predicting the Effects of Gene Deletion,* ACM SIGKDD Explorations Newsletter, Volume 4, Issue 2, pp. 101-103.

Determining causal and non-causal relationships in biomedical text by classifying verbs using a Naive Bayesian Classifier

Pieter van der Horn **Bart Bakker** **Gijs Geleijnse** **Jan Korst** **Sergei Kurkin**
Philips Research Laboratories
High Tech Campus 12a, 5656 AE Eindhoven, The Netherlands
{pieter.van.der.horn,bart.bakker,gijs.geleijnse,jan.korst,sergei.kurkin}@philips.com

1 Introduction

Since scientific journals are still the most important means of documenting biological findings, biomedical articles are the best source of information we have on protein-protein interactions. The mining of this information will provide us with specific knowledge of the presence and types of interactions, and the circumstances in which they occur.

There are various linguistic constructions that can describe a protein-protein interaction, but in this paper we will focus on subject-verb-object constructions. If a certain protein is mentioned in the subject of a sentence, and another protein in the object, we assume in this paper that some interaction is described between those proteins. The verb phrase that links the subject and object together plays an important role in this. However, there are a great many different verbs in the English language that can be used in a description of a protein-protein interaction. Since it is practically impossible to manually determine the specific biomedical meanings for all of these verbs, we try to determine these meanings automatically. We define two classes of protein-protein interactions, *causal* and *non-causal*, and using a Naive Bayesian Classifier, we predict for a given verb in which class it belongs. This process is a first step in automatically creating a useful network of interacting proteins out of information from biomedical journals.

2 Preprocessing

The protein-protein interactions we are interested in are described in the subject, the object and the interlinking verb phrase of a sentence. To determine which parts of the sentence make up this construction, we need to preprocess the sentence. For this, we use the Genia Chunker[1] to break the sentence into different chunks (in particular we are interested in noun phrases and verb phrases). We combine this information with the result of the Stanford Dependency Parser[2] to determine how these different chunks (phrases) are connected to each other.

3 Classification

The subject-verb-object construction can be schematically represented as follows:

[(state of) protein] [verb] [(state of) protein]

We make a distinction between two classes of verbs. One class describes a strict *causal relation* and the other covers all other types of meanings (*non-causal*). Table 1 shows some example verbs for the two classes.

Class	**Examples**
causal	*activate, inhibit, cause*
non-causal	*interact, require, bind*

Table 1: Two classes of verbs.

Since we leave out the information of the states of the proteins in this work, the first class covers positive, negative and neutral causal relations. The second class includes not just verbs that describe a correlation (*interact*), but also verbs such as *require*

[1] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/
[2] http://nlp.stanford.edu/downloads/lex-parser.shtml

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 112–113,
Columbus, Ohio, USA, June 2008.

and *bind* that describe a biologically important relationship, but not specifically a causal one.

We use a Naive Bayesian Classifier to estimate the probability $P(c_i|V)$ that a given verb belongs to a certain class. In the retrieved subject-verb-object constructions, such a verb V will occur a number of times, each time in combination with a specific ordered pair of proteins pp_j, one in the subject and one in the object. Each pair pp_j independently contributes to the estimation of $P(c_i|V)$.

$$V = \{pp_1, pp_2, ..., pp_n\} \quad (1)$$

$$P(c_i|V) = \frac{P(c_i) \cdot \prod_{j=1}^{n} P(pp_j|c_i)}{P(pp_1, pp_2, ..., pp_n)} \quad (2)$$

4 Experimental results

To test our approach, we retrieved a set of subject-verb-object relations from PubMed. We choose to test our approach on yeast proteins rather than e.g. human proteins to avoid Named Entity Recognition problems.

To get rid of any excess information, the verb phrases are normalized. We assume the last verb in the phrase to be the relevant verb and check the direction of the relation (active or passive form of that verb). Finally, the verb is stemmed. For those verbs that are in the passive form, the order of the protein pairs around it was reversed, and, for simplification, verb phrases that describe a negation were removed. More than one protein can occur in the subject and/or object, so we count each possible pair as an occurrence around the particular verb.

We used the 6 verbs as shown in Table 1 as a starting set to test the classifier. They represent the different types within each class, and of these it is clear they belong in that specific class. By using WordNet[3] we can augment this set. Table 1 shows the results of the different tests, using different parameter settings in WordNet to augment the training set ('l1' means recursive level 1, 's2' means WordNet senses 1 to 2, 'sa' means all WordNet senses are taken). It contains the number of verbs classified in the leave-one-out cross validation (*V*), the number of verbs that were correctly classified (*C*), the precision ($P = \frac{C}{V}$) and the probability Q that a random classifier would perform as good or better than this classifier, given by Equation 3

[3]http://wordnet.princeton.edu/

	V	*C*	*P*	*Q*
no WN	6	3	0.50	0.66
l1/s1	13	7	0.54	0.50
l1/s2	18	13	0.72	0.05
l1/sa	19	14	0.74	0.03
l2/s1	19	12	0.63	0.18
l2/s2	27	21	0.78	2.96E-3
l2/sa	55	32	0.58	0.14
l3/s1	26	20	0.77	4.68E-3
l3/s2	42	35	0.83	7.55E-6
l3/sa	73	43	0.59	0.08

Table 2: Results for different settings.

$$Q = \sum_{i=C}^{V} \binom{V}{i} p^i \cdot (1-p)^{V-i} \quad (3)$$

5 Conclusions and future work

Given an appropriate set of known verbs, we can predict the meanings of unknown verbs with reasonable confidence. This automatic prediction is very useful, since it is infeasible to manually determine the meanings of all possible verbs. We used two classes of verbs, making the distinction between relations that describe proteins *affecting* other proteins (*causal relation*) and any other relation (*non-causal relation*). Verbs like *require* and *bind* describe biologically distinct interactions however, and preferably should be put into classes separate from general correlations. We chose to use a two-way distinction as a first step however, which was still biologically relevant. In order to create a more detailed network of interacting proteins, one can take these other types into account as well.

Furthermore, it would be useful to separate the causal relationship into positive and negative relations. This specific distinction however is not just described in the connecting verb, but also in possible state descriptions in the noun phrases. Further research is necessary to extract these descriptions from the text. Finally, it would be useful to look at different syntactical constructions, other than just subject and object.

Statistical Term Profiling for Query Pattern Mining

Paul Buitelaar
DFKI GmbH
Language Technology Lab
Saarbrücken, Germany
paulb@dfki.de

Pinar Oezden Wennerberg, Sonja Zillner
Siemens AG
Knowledge Management CT IC 1
Munich, Germany
pinar.wennerberg.ext@siemens.com, sonja.zillner@siemens.com

1 Introduction

Through advanced technologies in clinical care and research, especially the rapid progress in imaging technologies, more and more medical imaging data and patient text data is generated by hospitals, pharmaceutical companies, and medical research. For enabling advanced access to clinical imaging and text data, it is relevant to know what kind of knowledge the clinician wants to know or the queries that clinicians are interested in. Through intensive interviews and discussions with radiologists and clinicians, we have learned that medical imaging data is analyzed - and hence queried – from three different perspectives, i.e. the *anatomic perspective* addressing the involved body parts, the *radiology-specific spatial perspective* describing the relationships of located anatomical regions to other anatomical parts, and the *disease perspective* distinguishing between normal and abnormal imaging features. Our aim is to establish query patterns reflecting those three perspectives that would typically be used by clinicians and radiologists to find patient-specific sets of relevant images.

The context of our work is in the Theseus-MEDICO[1] project on cross-modal image and information retrieval in the medical domain. The focus of the work reported here is on setting up Wikipedia-based corpora of human anatomy and radiology and on obtaining a statistical profile of concepts from three semantic knowledge resources with these corpora: the Foundational Model of Anatomy (FMA), the radiology lexicon RadLex, and a subset of the international classification of disease codes ICD-9 CM. Using this information, we intend to extract relations that are likely to occur between statistically relevant terms and the concepts they express.

The final goal of our work is to derive potential query patterns from the extracted set of relations that can be used in the MEDICO semantic-based image retrieval application. For example when re-staging head and neck lymphoma, clinicians and radiologists look for information and images that report on essential radiological patterns as *"an enlargement in the dimension of the lymph node in the neck"*. Therefore, within our approach, we aim at establishing hypotheses about possible user queries, i.e. the query patterns that reflect the three perspectives discussed above. Accordingly, an example query pattern might look like this:

[ANATOMICAL STRUCTURE]	*located_in*	[ANATOMICAL STRUCTURE]
	AND	
[[RADIOLOGY] IMAGE]Modality]	*is_about*	[ANATOMICAL STRUCTURE]
	AND	
[[RADIOLOGY IMAGE]Modality]	*shows_ symptom*	[DISEASE SYMPTOM]

Once an initial set of similar patterns has been established in this way, they will be evaluated by clinicians for their validity and relevance.

2 Corpora

A central aspect of the query pattern mining task is the statistical analysis of the FMA and RadLex terms in relevant text collections. In this way relevance scores can be assigned to terms that allow to investigate the most likely expressed (and hence queried) relations between them. For this purpose we need access to a representative corpus of texts that at the same time reflects the joint view of anatomy, spatial aspects of radiology and disease that we are targeting. Patient records would be our first choice, but due to strict anonymization requirements these are difficult to obtain. We therefore constructed a corpus based on the Wikipedia Categories Anatomy and Radiology. We then ran all text sections of each corpus through a part-of-speech tagger (Brants, 2000) to extract all nouns in the corpus and to compute a relevance score (chi-square) for each by comparing anatomy and radiology frequencies with those in the British Na-

[1] http://theseus-programm.de/scenarios/en/medico

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 114–115,
Columbus, Ohio, USA, June 2008.

tional Corpus. A next step will be to parse and annotate sentences with predicate-structure information, which may then be used for relation extraction along the lines of (Schutz and Buitelaar, 2005).

3 FMA Terms

The statistically most relevant FMA terms were identified on the basis of chi-square scores computed for nouns in each corpus. Single word terms in the FMA and occurring in the corpus correspond directly to the noun that the term is build up of (e.g. the noun 'ear' corresponds to the FMA term *ear*). In this case, the statistical relevance of the term is the chi-square score of the corresponding noun. In the case of multi-word terms occurring in the corpus, the statistical relevance is computed on the basis of the chi-square score for each constituting noun and/or adjective in the term, summed and normalized over the length of the term. Thus, the relevance value for *lymph node* is the summation of chi-square scores for 'lymph' and 'node' divided by 2. In order to take frequency in account, we further multiplied the summed relevance value by the frequency of the term. This assures that only frequently occurring terms are judged as relevant.

FMA Term	Freq.	Score	POS
lateral	464	338724,00	JJ
anterior	452	314721,00	JJ
artery	237	281961,00	NN
anterior spinal artery	2	219894,33	JJ JJ NN
lateral thoracic artery	2	217815,33	JJ JJ NN

Table 1: top FMA terms in anatomy corpus

FMA Term	Freq.	Score	POS
artery	65	6724,00	NN
coronary artery	17	5284,00	JJ NN
small bowel	11	4651,79	JJ NN
renal artery	3	4286,50	JJ NN
pulmonary artery	1	3974,50	JJ NN

Table 2: top FMA terms in radiology corpus

4 RADLEX Terms

Analogously, RadLex was used to identify the most relevant radiology terms. The most relevant RadLex terms are shown below. As with the FMA, the most relevant RadLex terms in the anatomy corpus are centered on "artery". In contrast, in the radiology corpus the RadLex relevance scores indeed point to a radiology profile:

RadLex Term	Freq.	Score	POS
lateral	464	338724,00	JJ
anterior	452	314721,00	JJ
artery	237	281961,00	NN
anterior spinal artery	2	219894,33	JJ JJ NN
lateral thoracic artery	2	217815,33	JJ JJ NN

Table 3: top RadLex terms in anatomy corpus

RadLex Term	Freq.	Score	POS
x-ray	253	81901,64	NN
imaging modality	6	58682,00	NN NN
volume imaging	1	57855,09	NN NN
molecular imaging	4	57850,00	JJ NN
mr imaging	9	57850,00	JJ NN

Table 4: topRadLex terms in radiology cor pus

5 Conclusions and Future Work

Using ICD-9 lymphoma terminology, we will derive a Pubmed-based corpus on lymphoma to analyse the context of the statistically top most relevant terms from the FMA and RadLex terminologies. In this way we will be able to identify relationships and eventually query patterns across the three dimensions of anatomy, radiology and lymphoma research.

Acknowledgments

This research has been supported in part by the THESEUS- MEDICO Project, which is funded by the German Federal Ministry of Economics and Technology under the grant number 01MQ07016.

References

Brants T. (2000). TnT - A Statistical Part-of-Speech Tagger. In: Proc. of the 6th ANLP Conference, Seattle, WA

Langlotz, CP. (2006). RadLex: A New Method for Indexing Online Educational Materials In: *Radiographics* 26, pp.1595-1597.

Rosse C. and J.L.V. Mejino Jr. (2003). A reference ontology for biomedical informatics: The Foundational Model of Anatomy. *Journal of Biomedical Informatics*, 36(6), pp. 478–500.

Schutz A., and P Buitelaar. (2005). RelExt: A Tool for-Relation Extraction in Ontology Extension In: *Proc. the 4th International Semantic Web Conference*, Galway, Ireland.

Using Language Models to Identify Language Impairment in Spanish-English Bilingual Children

Thamar Solorio
Department of Computer Science
The University of Texas at Dallas
tsolorio@hlt.utdallas.edu

Yang Liu
Department of Computer Science
The University of Texas at Dallas
yangl@hlt.utdallas.edu

1 Introduction

Children diagnosed with Specific Language Impairment (SLI) experience a delay in acquisition of certain language skills, with no evidence of hearing impediments, or other cognitive, behavioral, or overt neurological problems (Leonard, 1991; Paradis et al., 2005/6). Standardized tests, such as the Test for Early Grammatical Impairment, have shown to have great predictive value for assessing English speaking monolingual children. Diagnosing bilingual children with SLI is far more complicated due to the following factors: lack of standardized tests, lack of bilingual clinicians, and more importantly, the lack of a deep understanding of bilingualism and its implications on language disorders. In addition, bilingual children often exhibit code-switching patterns that will make the assessment task even more challenging. In this paper, we present preliminary results from using language models to help discriminating bilingual children with SLI from Typically-Developing (TD) bilingual children.

2 Our Approach

We believe that statistical inference can assist in the problem of accurately discriminating language patterns indicative of SLI. In this work, we use Language Models (LMs) for this task since they are a powerful statistical measure of language usage and have been successfully used to solve a variety of NLP problems, such as text classification, speech recognition, hand-writing recognition, augmentative communication for the disabled, and spelling error detection (Manning and Schütze, 1999). LMs estimate the probability of a word sequence $W = \langle w_1, ...w_k \rangle$ as follows (using the chain rule):

$$p(W) = \prod_{i=1}^{k} p(w_i|w_1, \ldots, w_{i-1})$$

which can be approximated using an N-gram as:

$$p(W) \approx \prod_{i=1}^{k} p(w_i|w_{i-N+1}, w_{i-N+2}, ..., w_{i-1})$$

Since in our problem we are interested in differentiating syntactic patterns, we will train the LMs on Part-of-Speech (POS) patterns instead of words. Using a 3-gram we have:

$$p(T) = \prod_{i=1}^{k} p(t_i|t_{i-2}, t_{i-1})$$

where $T = \langle t_1, t_2, ..., t_k \rangle$ is the sequence of POS tags assigned to the sequence of words W.

The intuition is that the language patterning of an SLI child will differ from those of TD children at two different levels: one is at the syntactic level, and the second one is at the interaction between both languages in patterns such as code-switching. Given that the tagset for each language is different, by using the POS tags we will incorporate into the model the syntactic structure together with the switch points across languages.

We train two LMs with the POS sequences: M_T, with data from the TD children and M_I, with data from the SLI bilingual children. Once both LMs are trained, then we can use them to make predictions over new speech samples of bilingual children. To determine whether an unobserved speech sample is likely to belong to a child suffering from SLI, we will measure the perplexity of the two LMs over the POS patterns of this new speech sample. We make the final decision using a threshold:

$$d(s) = \begin{cases} SLI & if\ (PP_T(s) - PP_I(s)) > 0 \\ TD & otherwise \end{cases}$$

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 116–117,
Columbus, Ohio, USA, June 2008.

where $PP_T(s)$ is the perplexity of the model M_T over the sample s, and $PP_I(s)$ is the perplexity of the model M_I over the same sample s. In other words, if the perplexity of the LM trained on syntactic patterns of children with SLI is smaller than that of the LM trained on POS patterns of TD children, then we will predict that the sample belongs to a child with SLI.

In a related work, (Roark et al., 2007) explored the use of cross entropy of LMs trained on POS tags as a measure of syntactic complexity. Their results were inconsistent across language tasks, which may be due to the meaning attached to cross entropy in this setting. Unlikely patterns are a deviation from what is expected; they are not necessarily complex or syntactically rich.

3 Preliminary Results

We empirically evaluated our approach using transcripts that were made available by a speech pathologist in our team. The TD samples were comprised of 5 males and 4 females between 48 and 72 months old. The children were identified as being bilingual by their parents, and according to parental report, these children live in homes where Spanish is spoken an average of 46.3% of the time. Language samples of SLI bilinguals were collected from children being served in the Speech and Hearing Clinic at UTEP. The samples are from two females aged 53 and 111 months. The clients were diagnosed with language impairment after diagnostic evaluations which were conducted in Spanish. The transcriptions were POS tagged with the bilingual tagger developed by (Solorio et al., 2008).

Table 1 shows the preliminary results using cross validation. With the decision threshold outlined above, out of the 9 TD children, the models were able to discriminate 7 as TD; from the 2 SLI children both were correctly identified as SLI. Although the results presented above are not conclusive due to the very small size corpora at hand, they look very promising. Stronger conclusions can be drawn once we collect more data.

Table 1: Perplexity and final output of the LMs for the discrimination of SLI and TD.

Sample	$PP_T(s)$	$PP_I(s)$	$d(s)$
TD_1	14.73	23.12	TD
TD_2	11.37	16.17	TD
TD_3	18.35	36.58	TD
TD_4	30.23	22.27	SLI
TD_5	9.42	15.50	TD
TD_6	17.37	36.75	TD
TD_7	20.32	33.19	TD
TD_8	16.40	24.47	TD
TD_9	24.35	23.71	SLI
SLI_1	20.21	19.10	SLI
SLI_2	19.70	12.43	SLI
average TD	18.06	25.75	TD
average SLI	19.95	15.76	SLI

4 Final Remarks

This paper presents very promising preliminary results on the use of LMs for discriminating patterns indicative of SLI in Spanish-English bilingual children. As more data becomes available, we expect to gather stronger evidence supporting our method. Our current efforts involve collecting more samples, as well as evaluating the accuracy of LMs on monolingual children with and without SLI.

Acknowledgements

Thanks to Bess Sirmon Fjordbak for her contribution to the project and the three anonymous reviewers for their useful comments.

References

L. B. Leonard. 1991. Specific language impairment as a clinical category. *Language, Speech, and Hearing Services in Schools*, 22:66–68.

C. D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

J. Paradis, M. Crago, and F. Genesee. 2005/6. Domain-general versus domain-specific accounts of specific language impairment: Evidence from bilingual childrens acquisition of object pronouns. *Language Acquisition*, 13:33–62.

B. Roark, M. Mitchell, and K. Hollingshead. 2007. Syntactic complexity measures for detecting mild cognitive impairment. In *BioNLP 2007: Biological, translational, and clinical language processing*, pages 1–8, Prague, June. ACL.

T. Solorio, Y. Liu, and B. Medina. 2008. Part-of-speech tagging English-Spanish code-switched text. *Submitted to Natural Language Engineering*.

Raising the Compatibility of Heterogeneous Annotations: A Case Study on Protein Mention Recognition

Yue Wang* **Kazuhiro Yoshida*** **Jin-Dong Kim*** **Rune Sætre*** **Jun'ichi Tsujii*†‡**
*Department of Computer Science, University of Tokyo
†School of Informatics, University of Manchester
‡National Center for Text Mining
Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033 JAPAN
{wangyue, kyoshida, jdkim, rune.saetre, tsujii}@is.s.u-tokyo.ac.jp

Abstract

While there are several corpora which claim to have annotations for protein references, the heterogeneity between the annotations is recognized as an obstacle to develop expensive resources in a synergistic way. Here we present a series of experimental results which show the differences of protein mention annotations made to two corpora, GENIA and AImed.

1 Introduction

There are several well-known corpora with protein mention annotations. It is a natural request to benefit from the existing annotations, but the heterogeneity of the annotations remains an obstacle. The heterogeneity is caused by different definitions of "protein", annotation conventions, and so on.

It is clear that by raising the compatibility of annotations, we can reduce the performance degradation caused by the heterogeneity of annotations.

In this work, we design several experiments to observe the effect of removing or relaxing the heterogeneity between the annotations in two corpora. The experimental results show that if we understand where the difference is, we can raise the compatibility of the heterogeneous annotations by removing the difference.

2 Corpora and protein mention recognizer

We used two corpora: the GENIA corpus (Kim et al., 2003), and the AImed corpus (Bunescu and Mooney, 2006). There are 2,000 MEDLINE abstracts and 93,293 entities in the GENIA corpus. The annotation is dependent on a small taxonomy of 36 classes. The AImed corpus consists of 225 MEDLINE abstracts, and there are 4,084 protein references.

Our protein mention recognizer is a Maximum Entropy Markov Model (MEMM) n-best tagger.

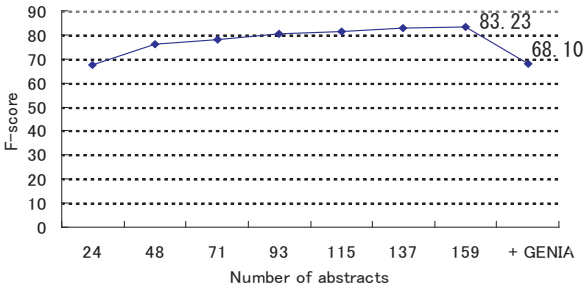

Figure 1: The learning curve according to the F-score

3 The effect of the inconsistency

We did two experiments in order to characterize the following two assumptions. First, we can improve the performance by increasing the size of the training data set. Secondly, the system performance will drop when more inconsistent annotations are introduced into the training data set.

In these two experiments, for the training, we used the AImed corpus and the AImed corpus plus the GENIA protein annotations, respectively. We conducted the evaluation on the AImed corpus.

The learning curve drawn from the results of the two mentioned experiments is shown in Figure 1. We can see that the learning curve is still increasing

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 118–119,
Columbus, Ohio, USA, June 2008.

Subcategory	Recall	Precision	F-score
Family_or_group	12.94	3.86	5.94
Domain_or_region	15.74	0.57	1.11
Molecule	48.80	34.43	40.37
Substructure	0.00	0.00	0.00
Subunit	65.36	3.38	6.43
Complex	13.43	0.98	1.83
ETC	14.29	0.03	0.07

Table 1: The experimental results on seven subclasses.

AImed + Subcategory	Criterion	F-score
Molecule+Subunit	Exact	64.72
	Left	69.48
	Right	67.64
Molecule+Subunit+Complex	Exact	63.76
	Left	72.77
	Right	67.60

Table 2: The experimental results on three subclasses.

when we used up all the training portions from the AImed corpus. Even though the rate of the improvement is slow, we would expect a further improvement if we could add more training data in a large scale, e.g. the GENIA corpus is 10 times bigger than the AImed corpus. But when we added the protein annotations in the GENIA corpus to the training data set, we witnessed a drastic degradation in the performance. We assume that the degradation is caused by the heterogeneity of the protein annotations in these two corpora, and we further assume that if the heterogeneity could be eliminated, the learning curve would go back to an increasing state.

4 Raising the compatibility

Although both corpora include protein mention annotations, the target task is different. *GENIA concerns all the protein-mentioning terms, while AImed focuses only on the references of individual proteins*. In the GENIA corpus, besides the 36 classes, some subclasses are also included. In the case with the protein class, there are seven subclasses: family_or_group, domain_or_region, molecule, substructure, subunit, complex, etc. Further, in the AImed corpus, protein/gene families are not tagged, only protein molecules are tagged.

We conducted an experiment to verify what we found from the documentation of the two corpora. We trained our tagger using the AImed corpus, and evaluated it on the GENIA corpus. Each time, we assumed only the annotation of one protein subclass in the GENIA corpus as the "gold" annotation. Table 1 shows the experimental results.

The experimental results clearly supported the documented scope of the protein annotation in GENIA and AImed: The protein mention recognizer trained with AImed best recognized the GENIA annotation instances of Protein_molecules among all subclasses, and the performance of recognizing Protein_family_or_group instances was very poor.

We therefore have a hypothesis: if we unite the GENIA annotations of Protein_molecule, Protein_subunit, and Protein_complex with the AImed corpus, and we use this united corpus to train our tagger, we can improve the performance of our tagger on the AImed corpus. Table 2 shows our experimental results based on this hypothesis. It can be seen from the result that, if we assume that the upper bound of the F-score of this approach is near to 83.23%, we reduced the incompatibility of the two corpora by 30%. The reduction was obtained by understanding the difference of the protein annotations made to the corpora.

5 Conclusion

We implemented several experiments in order to remove the negative influence of the disagreements between two corpora. Our objective is to raise the compatibility of heterogeneous annotations. Some simple experiments partly revealed where the heterogeneity between the protein mention annotations in GENIA and AImed is. More qualitative and quantitative analysis will be done to identify the remaining heterogeneity.

References

Razvan Bunescu and Raymond Mooney. 2006. Subsequence Kernels for Relation Extraction. *Advances in Neural Information Processing Systems*, 18:171–178.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi and Jun'ichi Tsujii. 2003. GENIA Corpus - a Semantically Annotated Corpus for Bio-textmining. *Bioinformatics*, 19(Suppl. 1):i180–i182.

Adaptive Information Extraction for Complex Biomedical Tasks

Donghui Feng **Gully Burns** **Eduard Hovy**
Information Sciences Insitute
University of Southern California
Marina del Rey, CA, 90292
{donghui, burns, hovy}@isi.edu

Abstract

Biomedical information extraction tasks are often more complex and contain uncertainty at each step during problem solving processes. We present an adaptive information extraction framework and demonstrate how to explore uncertainty using feedback integration.

1 Adaptive Information Extraction

Biomedical information extraction (IE) tasks are often more complex and contain uncertainty at each step during problem solving processes.

When in the first place the desired information is not easy to define and to annotate (even by humans), iterative IE cycles are to be expected. There might be gaps between the domain knowledge representation and computer processing ability. Domain knowledge might be hard to represent in a clear format easy for computers to process. Computer scientists may need time to understand the inherent characteristics of domain problems so as to find effective approaches to solve them. All these issues mandate a more expressive IE process.

In these situations, the traditional, straightforward, and one-pass problem-solving procedure, consisting of definition-learning-testing, is no longer adequate for the solution.

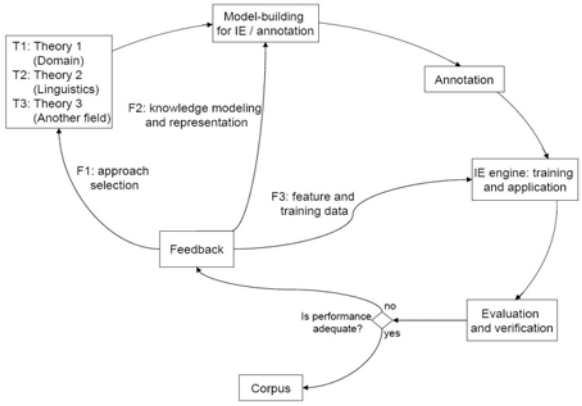

Figure 1. Adaptive information extraction.

For more complex tasks requiring iterative cycles, an adaptive and extended IE framework has not yet been fully defined although variants have been explored. We describe an adaptive IE framework to characterize the activities involved in complex IE tasks. Figure 1 depicts the adaptive information extraction framework.

This procedure emphasizes one important adaptive step between the learning and application phases. If the IE result is not adequate, some adaptations are required:

Our study focuses on extracting tract-tracing experiments (Swanson, 2004) from neuroscience articles. The goal of tract-tracing experiment is to chart the interconnectivity of the brain by injecting tracer chemicals into a region of the brain and then identifying corresponding labeled regions where the tracer is transported to (Burns *et al*., 2007). Our work is performed in the context of NeuroScholar[1], a project that aims to develop a Knowledge Base Management System to benefit neuroscience research.

We show how this new framework evolves to meet the demands of the more complex scenario of biomedical text mining.

2 Feedback Integration

This task requires finding the knowledge describing one or more experiments within an article as well as identifying desired fields within individual sentences. Significant complexity arises from the presence of a variable number of records (experiments) in a single research article --- anywhere from one to many.

Experiment	
tracerChemical	null
injectionLocation	the contralateral AVCN
labelingLocation	the DCN
labelingDescription	no labelled cells

Table 1. An example tract-tracing experiment.

Table 1 provides an example of a tract-tracing experiment. In this experiment, when the tracer was injected into the injection location “the contralateral AVCN”, “no labeled cells” was found in the labeling location “the DCN”.

For sentence level fields labeling, the performance of F1 score is around 0.79 (Feng *et al*., 2008).

[1] http://www.neuroscholar.org/

BioNLP 2008: Current Trends in Biomedical Natural Language Processing, pages 120–121,
Columbus, Ohio, USA, June 2008.

We here show how the adaptive information extraction framework is applied to labeling individual sentences. Please see Feng *et al*. (2007) for the details of segmenting data records.

2.1 Choosing Learning Approach via F1

A natural way to label sentences is to obtain (by hand or learning) patterns characterizing each field (Feng *et al*., 2006; Ravichandran and Hovy, 2002). We tried to annotate field values for the biomedical data, but we found few intuitive clues that rich surface text patterns could be learned with this corpus.

This insight, Feedback F1, caused us to give up the idea of learning surface text patterns as usual, and switch to the Conditional Random Fields (CRF) (Lafferty *et al*., 2001) for labeling sentences instead. In contrast to fixed-order patterns, the CRF model provides a compact way to integrate different types of features for sequential labeling problems and can reach state-of-the-art level performance.

2.2 Determining Knowledge Schema via F2

In the first place, it is not clear what granularity of knowledge/information can be extracted from text and whether the knowledge representation is suitable for computer processing. We tried a series of approaches, using different levels of granularity and description, in order to obtain formulation suitable for IE. Figure 2 represents the evolution of the knowledge schema in our repeated activities.

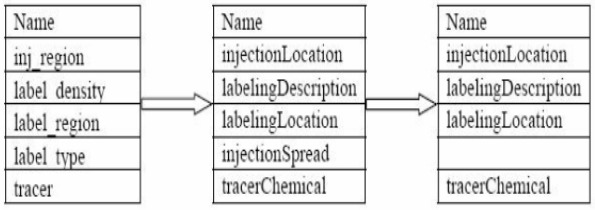

Figure 2. Knowledge schema evolution.

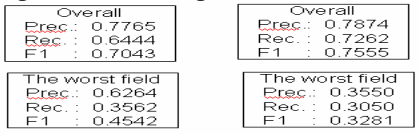

Figure 3. System performance at stage 1 and 2.

We initially started with the schema in the leftmost column but our pilot study showed that some fields, for example, "label_type", had too many variations in text description, making it very hard for CRF to learn clues about it. We then switched to the second schema but ended up seeing that the field "injectionSpread" needed more domain knowledge and was therefore not able to be learned by the systems. The last column is the final schema after those pilot studies. Figure 3 shows system performance (overall and the worst field) corresponding to the first and the second representation schemas.

2.3 Exploring Features via F3

To train CRF sentence labeling systems, it is vital to decide what features to use and how to prepare those features. Through the cycle of Feedback F3, we explored five categories of features and their combinations to determine the best features for optimal system performance. Table 2 shows system performance with different feature combinations.

System Features	Prec.	Recall	F_Score
Baseline	0.4067	0.1761	0.2458
Lexicon	0.5998	0.3734	0.4602
Lexicon + Surface Words	0.7663	0.7302	0.7478
Lexicon + Surface Words + Context Window	0.7717	0.7279	0.7491
Lexicon + Surface Words + Context Window + Window Words	0.8076	0.7451	0.7751
Lexicon + Surface Words + Context Window + Window Words + Dependency Features	**0.7991**	**0.7828**	**0.7909**

Table 2. Precision, Recall, and F_Score for labeling.

Please see Feng *et al*. (2008) for the details of the sentence level extraction and feature preparation,

3 Conclusions

In this paper, we have shown an adaptive information extraction framework for complex biomedical tasks. Using the iterative development cycle, we have been able to explore uncertainty at different levels using feedback integration.

References

Burns, G., Feng, D., and Hovy, E.H. 2007. Intelligent Approaches to Mining the Primary Research Literature: Techniques, Systems, and Examples. Book Chapter in *Computational Intelligence in Bioinformatics*, Springer-Verlag, Germany.

Feng, D., Burns, G., and Hovy, E.H. 2007. Extracting Data Records from Unstructured Biomedical Full Text. In *Proc. of EMNLP 2007*.

Feng, D., Burns, G., Zhu, J., and Hovy, E.H. 2008. Towards Automated Semantic Analysis on Biomedical Research Articles. In *Proc. of IJCNLP-2008*. Poster Paper.

Feng, D., Ravichandran, D., and Hovy, E.H. 2006. Mining and reranking for answering biographical queries on the web. In *Proc. of AAAI-2006*. pp. 1283-1288.

Lafferty, J., McCallum, A. and Pereira, F. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML-2001*.

Ravichandran, D. and Hovy, E.H. 2002. Learning surface text patterns for a question answering system. In *Proceedings of ACL-2002*.

Swanson, L.W. 2004. *Brain maps: structure of the rat brain*. 3rd edition, Elsevier Academic Press.

Author Index

Airola, Antti, 1
Ananiadou, Sophia, 30, 63
Arszman Lavanier, Sarah, 96

Bakker, Bart, 112
Bergler, Sabine, 46
Björne, Jari, 1
Bloodgood, Michael, 104
Buitelaar, Paul, 114
Burns, Gully, 120

Carazo, José María, 108
Chagoyen, Monica, 108
Chapman, Wendy, 106
Chun, Hong-Woo, 90
Claveau, Vincent, 88
Combs, Jennifer, 96
Copestake, Ann, 54
Corbett, Peter, 54
Csirik, János, 38

Farkas, Richárd, 38
Feng, Donghui, 120
Frid, Nadya, 92
Frunza, Oana, 110

Gaizauskas, Robert, 10, 80
Geleijnse, Gijs, 112
Ginter, Filip, 1
Gojobori, Takashi, 90
Grupp-Phelan, Jacqueline, 96
Guo, Yinkun, 80

Haddow, Barry, 19
Harkema, Henk, 106
He, Yulan, 98
Hepple, Mark, 10
Hishiki, Teruyoshi, 90
Hovy, Eduard, 120

Imanishi, Tadashi, 90
Inkpen, Diana, 110

Joshi, Aravind, 92

Kilicoglu, Halil, 46
Kim, Jin-Dong, 90, 118
Kipper-Schuler, Karin, 94
Korst, Jan, 112
Kowatch, Robert, 96
Kurkin, Sergei, 112

Lee, Alan, 92
Li, Dingcheng, 94
Liu, Yang, 116

Martinez, David, 80
Martínez, Paloma, 100
Mattews, Michael, 71
Matykiewicz, Pawel, 96
McNaught, John, 63
McRoy, Susan, 92
Mowery, Danielle, 106

Neveol, Aurelie, 88
Neves, Mariana, 108

Oezden Wennerberg, Pinar, 114

Pahikkala, Tapio, 1
Pascual-Montano, Alberto, 108
Patrick, Jon, 102
Pestian, John, 96
Prasad, Rashmi, 92
Pyysalo, Sampo, 1

Roberts, Angus, 10

Saetre, Rune, 118
Saichi, Naomi, 90
Salakoski, Tapio, 1

Samy, Doaa, 100
Sasaki, Yutaka, 63
Savova, Guergana, 94
Schmidt, Carl J, 28
Segura-Bedmar, Isabel, 100
Shooshan, Sonya, 88
Solorio, Thamar, 116
Stevenson, Mark, 80
Szarvas, György, 38

Takagi, Toshihisa, 90
Tanaka, Masayuki, 90
Tsujii, Jun'ichi, 30, 90, 118
Tsuruoka, Yoshimasa, 30, 63
Tudor, Catalina O, 28

van der Horn, Pieter, 112
Vijay-Shanker, K, 28, 104
Vincze, Veronika, 38

Wang, Xinglong, 71
Wang, Yefeng, 102
Wang, Yue, 118

Yamasaki, Chisato, 90
Yoshida, Kazuhiro, 118
Yu, Hong, 92

Zhou, Deyu, 98
Zillner, Sonja, 114