Integrated Computer-Aided Engineering 16 (2009) 339–352
DOI 10.3233/ICA-2009-0321
IOS Press

Using conditional random fields for result identification in biomedical abstracts

Ryan T.K. Lin[a], Hong-Jie Dai[a,b], Yue-Yang Bow[a], Justin Liang-Te Chiu[c] and Richard Tzong-Han Tsai[d*]
[a]*Institute of Information Science, Academia Sinica, Taipei, Taiwan*
[b]*Department of Computer Science, National Tsing-Hua University, Hsinchu, Taiwan*
[c]*Department of Computer Science & Engineering, National Taiwan University, Taipei, Taiwan*
[d]*Department of Computer Science & Engineering, Yuan Ze University, Chung-Li, Taiwan*

Abstract. The abstracts of biomedical papers usually contain three sections: objective, methods, and results-conclusion. The results-conclusion section is the most important because it usually describes the main contribution of a paper. Unfortunately, not all biomedical journals follow this three-section format. In this paper, we propose a machine learning (ML) based approach to automatically identify the results-conclusion section. The results-conclusion section identification problem is formulated as a sequence labeling task. Four feature sets, including Position, Named Entity, Tense, and Word Frequency, are employed with Conditional Random Fields (CRFs) as the underlying ML model. The experiment results show that the proposed approach can achieve F-measure, precision, and recall of 97.08%, 96.63% and 97.53%, respectively.

Keywords: Result identification, sequence labeling, conditional random fields

1. Introduction

Integrated computer-aided tools are widely developed and demonstrated their effectiveness in improving the working efficiency in many domains [1,3,7,20, 35,44]. In the biomedical domain, the phenomenal growth in biomedical literature poses a major problem for biologists. In recent years, a range of text-mining applications have been developed to improve access to knowledge for biologists and database curators [6, 11,21]. These applications perform functions like recognizing named entities (NEs) or identifying the relationships between them from an entire abstract without distinguishing the introduction from the methodology, the results or the conclusion. However, in the biomedical field, the results-conclusion section of an abstract usually describes the main contribution or new finding of a paper. For example, Fig. 1 shows a biomedical abstract (PubMed ID "15097232") comprised of four sections: objective, methods, results, and conclusion. The sentence in the results section, "*Genetic variation in GRK4gamma was associated with HT in the subjects studied*", summarizes the true contribution of the article. Therefore, distinguishing the results-conclusion section from the other parts of an abstract could provide detailed information on the article's key content. (For convenience, we refer to the results-conclusion section as the "results section" hereafter.)

In this work, we use the Conditional Random Fields (CRFs) [27] machine learning (ML) algorithm to identify the result section from other paragraphs because CRF is one of the best known model in solving sequential labeling tasks such as named entity recognition (NER) [39] and part of speech tagging [33], etc. Four feature sets, Position, Named Entity (NE), Tense, and Word Frequency (WF), are proposed. Because there are not any openly available section identification corpora, we generated our training and test data from a controlled source, hypertension-gene relation articles,

*Corresponding author: Dr. Richard Tzong-Han Tsai, Assistant Professor, Department of Computer Science and Engineering, Yuan Ze University, Chung-Li, Taiwan. Tel.: +886 3 4638800x3004; Fax: +886 3 4638850; E-mail: thtsai@saturn.yzu.edu.tw.

ISSN 1069-2509/09/$17.00

OBJECTIVE:
To perform association studies of polymorphisms of the potential candidate essential hypertension (HT) genes GRK4, PTP1B and HSD3B1.

METHODS:
Subjects consisted of 168 unrelated, Caucasian essential hypertensive (HT) patients and 312 normotensive (NT) controls. Biological power was increased by ensuring subjects in each group had parents with the same blood pressure (BP) status as theirs. Three GRK4gamma variants (R65L, A142V and A486V), one HSD3B1 variant (T<---C Leu) and one PTP1B variant (1484insG) were genotyped by polymerase chain reaction and restriction enzyme digestion or by homogenous MassEXTEND Assay.

RESULTS:
The V allele of the A486V variant of GRK4gamma, but not the R65L or A142V variants, showed an association with HT (P = 0.02). The V allele was also associated with an elevation in systolic blood pressure (SBP) (P = 0.002). Although the L65 and the V142 alleles tracked with elevation in diastolic (DBP), this was seen only in male HTs (P = 0.009; P = 0.002, respectively). Haplotype frequencies differed between the HT and NT groups, particularly for the R, V, V haplotype combination of R65L, A142V and A486V, respectively. Neither of the HSD3B1 or PTP1B variants were associated with HT.

CONCLUSION:
Genetic variation in GRK4gamma was associated with HT in the subjects studied.

Fig. 1. An example of a biomedical abstract.

to evaluate the proposed methods. Our in-lab biologists with the hypertension research background helped us check the controlled articles to examine the result section boundary. Several experiments on the corpus are conducted to evaluate the proposed feature sets and the combinations of them.

The remainder of this paper is organized as follows: Section 2 provides an overview of the state of the art. In Section 3, the CRFs model and our proposed feature sets are depicted. Section 4 reports the experiment results. Section 5 discusses feature set combinations and provides detailed analyses. A comparison between our proposed CRFs-based approaches with others is also described. Finally, Section 6 summarizes our conclusions.

2. Related work

In the light of rhetorical structure theory [18], clauses in text are relevant to one another through relations such as Background, Elaboration, and Contrast. These rhetorical relations when identified could be helpful for summarization, information retrieval, information extraction, and question answering. In natural language processing (NLP), researchers have undertaken to recognize rhetorical relations using manually crafted and statistical techniques [16,32].

[19,24,34] have claimed that abstracts across scientific disciplines including the biomedical field follow consistent rhetorical roles or "argumentative moves" (e.g. Problem, Solution, Evaluation, and Conclusion). Teufel and Moens [36] have proposed a strategy for summarization by classifying sentences from scientific texts into seven rhetorical categories. Extracted sentences could be concatenated for automated user-tailored summaries.

Since then, several result identification approaches have been proposed in recent years. In 2003, Shimbo et al. [29] present an experimental text retrieval system to facilitate search in the MEDLINE database. A unique feature of the system is that the user can search not only throughout the whole abstract text, but also from the limited "sections" in the text. For this purpose, they exploit the "structured" abstracts contained in the MEDLINE database in which sections are explicitly marked by the headings. These abstracts provide training data for constructing sentence classifiers that are used to section unstructured abstracts, in which explicit section headings are missing. They used support vector machine (SVM) [12] to classify sentences represented by bigrams, and contextual information and reported 91.9% accuracy.

Two years later, Yamamoto et al. [43] also used SVM classifiers with various novel features, such as subject-verb, verb tense, relative sentence location, and sentence score (i.e., the average TF-IDF score of constituent words) features and trained each of them for a different rhetorical status on structured abstracts. A structured abstract is one that has labels indicating rhetorical statuses of the sentences, while an unstructured abstract does not. The classifiers were tested on

both structured and unstructured abstracts. The former were randomly obtained from the MEDLINE database and the latter were manually labeled by humans. Their method achieved F-measure of 87.2% and 89.8% for result and conclusion (structured abstract) and F-measure of 81.8% and 87.4% for result and conclusion (unstructured abstract).

Recently, Ruch et al. [23] reported on the construction of a categorizer, which classifies sentences of biomedical abstracts into a four-class argumentative model. The system is based on a set of Bayesian learners trained on automatically acquired corpora and augmented with distributional heuristics. Feature weighting was optimal with DF-thresholding. For the CONCLUSION class, which has been reported to contain more highly informative contents than other sentences, they obtain an F-score of 85%.

Lin et al. [15] describes experiments with generative models for analyzing the discourse structure of medical abstracts, which generally follow the pattern of "introduction", "methods", "results", and "conclusions". They demonstrate that Hidden Markov Models [45] are capable of accurately capturing the structure of such texts, and can achieve classification accuracy comparable to that of discriminative techniques. Kneser-Ney discounting and Katz backoff are utilized to generate bigram language models for each section and achieved an F-measure of 89.8% for the result section and 89.7% for the conclusion section.

Wu et al. [42] introduces a method for computational analysis of move structures in abstracts of research articles. They proposed a six-step learning process to train a collocation classifier and exploited the HMM to tag sentences. Their system achieved 80.54% precision.

3. Method

3.1. Formulation

We transformed the result identification problem into a sentence-by-sentence sequential labeling task [2]. Each sentence in an abstract is regarded as a token. Each token is associated with a tag that indicates the section boundary. We adopt the IOB2 format, which has been proven to be the most appropriate format for sequence tagging problems [26]. Therefore, three tags are used to determine whether or not the sentence belongs to the beginning (B), the inside/ending (I) or outside (O) of the result section (RS), that is, *B-RS*, *I-RS*, and O. For a given abstract, the problem can then be formulated as a problem of sequentially assigning one of three tags to each sentence.

3.2. Conditional random fields

CRFs are undirected graphical models trained to maximize a conditional probability [13]. A linear-chain CRF with parameters $\Lambda = \{\lambda_1, \lambda_2, \ldots\}$ defines a conditional probability for a state sequence $\mathbf{y} = y_1 \ldots y_T$ given an input sequence $\mathbf{x} = x_1 \ldots x_T$ as

$$P_\Lambda(y|x) = \frac{1}{Z_x} \exp\left(\sum_{t=1}^{T}\sum_{k} \lambda_k f_k(y_{t-1}, y_t, x, t)\right)$$

where Z_x is the normalization that makes the probability of all state sequences sum to one; $f_k(y_{t-1}, y_t, \mathbf{x}, t)$ is usually a binary-valued feature function, λ_k is its weight, t indicates the token's position in the sequence, and T stands for the length of the sequence. The feature function can measure any aspect of a state transition, $y_{t-1} \rightarrow y_t$, and the entire observation sequence **x** centered at the current time step t. For example, a feature function might have the value 1 when y_{t-1} is the state *B-RC*, y_t is the state *I-RC*, and x_t is 1st position. Large positive values for λ_k indicate a preference for such an event; large negative values mean that the event is unlikely.

The most probable label sequence for **x**,

$$y^* = \arg\max_y P_\Lambda(y|x),$$

can be efficiently determined using the Viterbi algorithm [22].

The parameters can be estimated by maximizing the conditional probability of a set of label sequences, each given their corresponding input sequences. The log-likelihood of a training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\colon i = 1, \ldots, M\}$ is written as:

$$\begin{aligned} L_\Lambda &= \sum_i \log P_\Lambda(y_i|x_i) \\ &= \sum_i \left(\sum_{t=1}^{T}\sum_{k} \lambda_k f_k(y_{t-1}, y_t, x, t) - \log Z_{x_i}\right). \end{aligned}$$

To optimize the parameters in CRFs, a quasi-Newton gradient-climber BFGS [28] is used.

3.3. Feature set

In this section, four proposed features including Position, Named Entity (NE), Tense, and Word Frequency (WF) are described in detail.

3.3.1. Position feature

Because the results section is usually located at the end of the abstract, the relative position of a sentence in an abstract provides useful information to determine whether it belongs to the results section. We use the following equation to calculate the relative position of a sentence:

$$\text{relative}_{\text{position}}(s, S) = 10 \left[\frac{s}{S} \right]$$

where s is the sentence's position in the abstract, and S is the total number of sentences in the abstract. The relative position is an integer, ranging from 1 to 10. Since our CRF model only allows binary features, we correspond one value to one feature. Therefore, we have 10 position features. If a sentence's relative position is 5, then the position feature corresponding to 5 will be enabled, while the other position features will be disabled.

3.3.2. NE feature

Since the title can be treated as a summary of an abstract, it may contribute some information related to the result section. NEs in the title are a kind of such information. Therefore, we firstly constructed a CRFs-based NE recognizer to complete such work. In our NER approach, very similar to the approach described in Section 3.1, the first step is to transform the original NE annotation into a token/tag format. Each word in a sentence is regarded as a token, and each token is associated with a tag that indicates the location of the token within the NE, for example, B, and I. These two tags denote respectively the start token and the following token of an NE. In addition, we also use the tag O to indicate that a token is not part of an NE. For example, the following annotated phrase in XML format:

> "<Gene> IL-2 gene </Gene> expression, <Gene> CD28 </Gene>, and <Gene> NF- kappa B </Gene>"

is transformed to the IOB2 format:

> "IL-2/B gene/I expression/O, /O CD28/B, /O and/O NF- kappa/ B/I".

The following section describes the numerical normalization as well as global pattern-based correction post-processing for our NE recognizer. The further information about our NER approach including the detail feature sets can be seen at [5].

3.3.2.1. NER: Numerical normalization

According to our observation, some proteins or genes of the same family usually differ in their numerical parts. For example, AKT-2 and AKT-3 belong to the same family – AKT. Therefore, we normalize all numerals into one. For example, both AKT-2 and AKT-3 are normalized to AKT-1. We name this approach as numerical normalization. The advantages of this approach include: (1) the number of features can be substantially reduced; (2) it is possible to transform unseen features into seen features; and (3) feature weights can be estimated more accurately. Take the gene names IL2, IL3, IL4, and IL5 for example. IL2, IL3, IL4 are in the training set, but IL5 is not. If we apply numerical normalization to these terms, they will all be normalized to IL1. Therefore, the number of features corresponding to the first three terms is reduced to a minimum of 1/3. Since IL5 and IL1 are treated alike and share the same weight, this unseen feature becomes a seen feature. According to our analysis, normalization generally increases overall Bio-NER accuracy. Suppose IL2 is annotated as "gene" three times, IL3 is annotated as "gene" six times, and IL4 is annotated as "gene" once and as "compound" once. The annotation of IL4 may confuse machine learning models. After numerical normalization, however, the first three terms are annotated as "gene" ten times and as "compound" only once. Therefore, the feature weights can be correctly estimated.

3.3.2.2. NER: Pattern extraction

The CRFs model only uses the information in the limited context window. It may fail if there are dependencies beyond the context window, for example, an NE may depend on the previous or next NE, or words among these NEs. To alleviate these problems, we apply global patterns composed of NEs and surrounding words. The first step in creating global patterns is applying the aforementioned numerical normalization to all sentences in the training sets. For each pair of sentences in the training set, we apply the Smith-Waterman local alignment algorithm [31] to find the longest common string, which is then added to the candidate pattern pool. During the alignment process, for each position, either of the two inputs that share the same word or NE tag can be counted as a match. The similarity function used in the Smith-Waterman algorithm is:

$$Sim(x, y) = \max \begin{cases} 1, x = y \\ 1, NE(x) = NE(y) \\ 0, otherwise \end{cases}$$

Table 1
Sections and their corresponding tenses

Section	Tense
Background information	Present tense
Principal activity	Past tense/Present perfect tense
Methodology	Past tense
Results	Past tense
Conclusions	Present tense/Tentative verbs/Modal auxiliaries

where x and y referred to any two compared tokens from the first and second input sentences, respectively. The similarity of two inputs is calculated by the Smith-Waterman algorithm based on a token-level similarity function [37].

Following example illustrates how patterns are extracted from a sentence pair in the training set. Given the following two tagged sentences:

... chemical/O interactions/O that/O **inhibit**/O butyrylcholinesterase/***B*** **and**/O ...

and

... combinations/O of/O chemicals/O that/O **inhibit**/O butyrylcholinesterase/***B*** **and**/O ...

, we will generate the "**inhibit** <**NE**> **and**" pattern. Here, we put the aligned words and tags in bold font. The first and last tokens in a pattern are constrained to be words, sentence beginning or ending symbols. The extracted patterns are composed of a headword, NE type and a tail-word–for example, "headword <NE type> tail-word." To test its effectiveness, each pattern is applied to the BioCreAtIvE II Gene Mention data set [30] to correct the NE tags of all sentences. If the pattern's error ratio exceeds a certain threshold, τ, it is filtered out.

3.3.2.3. NE feature for result identification

After constructing the NER tagger, two NE features, f_{NE} and f_{bNE}, are designed for result identification. The f_{NE} feature represents how many NEs co-occur in both the title and the sentence. The formal definition of f_{NE} can be defined as follows:

$$f_{NE}(\{NE_s\}, \{NE_{title}\}) =$$
$$\begin{cases} |\{NE_s\} \cap \{NE_{title}\}|, \{NE_s\} \cap \{NE_{title}\} \neq \phi \\ 0, \{NE_s\} \cap \{NE_{title}\} = \phi \end{cases},$$

where $\{NE_s\}$ is the set of NEs in the current sentence, and $\{NE_{title}\}$ is the set of NEs in the title. f_{bNE} is a special case of f_{NE}, that binarizes the value of f_{NE} to 1 or 0.

For example, if "P53" exists in both the title and current sentence twice, the value of f_{NE} is 2, but the value of f_{bNE} is 1.

Table 2
Part-of-speech patterns for tenses

Tense	Part-of-speech pattern
Present tense	1) VB
	2) VBZ
	3) VBP
Present perfect tense	4) VBZ -> VBN -> VBN
	5) VBZ -> VBN
	6) VBP -> VBN
	7) VBP -> VBN -> VBN
Past tense	8) VBD
Past perfect tense	9) VBD -> VBN
	10) VBD -> VBN -> VBN
Tentative verbs/Modal auxiliaries	11) MD -> VB

3.3.3. Tense feature

Weissberg and Buker [41] suggested that an abstract has five important sections: "Background information", "Principal activity", "Methodology", "Results", and "Conclusions". "Background information" and "Principal activity" belong to the "Objective" section; therefore, in general an abstract has four sections [10], namely: "objective", "method", "result", and "conclusion". Table 1 lists each section with the corresponding tense described by Weissberg and Buker [41].The "Background information" section, for example, is written in the present tense. We follow this writing convention in the design of the "Tense" feature.

The feature is generated by the following steps. First, we utilize GENIATagger [40] to produce part-of-speech (POS) patterns for each sentence in the abstract. Second, we arrange the POS patterns according to the tenses shown in Table 2. For instance, the POS pattern of the "Past perfect tense" is "VBD->VBN", where VBD refers to a past tense verb, and VBN refers to a past participle. Finally, for different tense of a sentence, including the "present tense", "past tense", "present perfect tense", "past perfect tense", or "tentative verbs/modal auxiliaries", we assign a feature value 1, 2, 3, 4, or 5 to represent them. If the tense doesn't belong to the aforementioned tenses, the value 0 is assigned.

3.3.4. WF feature

The WF feature is designed for evaluating the importance of word unigrams or bigrams in the result section. Two sub-features, the word unigram frequen-

Table 3
Important word unigrams

Word Unigram	Only in Result (times)	In abstract (times)	Ratio (%)
Conclude	99	100	99
significantly	3514	3550	98.99
0.005	97	98	98.98
0.001	863	872	98.97
0.0001	303	307	98.70
0.01	109	111	98.20
Ci	1046	1070	97.76
0.04	87	89	97.75
0.05	1036	1062	97.55
0.01	643	660	97.42

Table 4
Important word bigrams

Word Bigram	Only in Result (times)	In abstract (times)	Ratio (%)
% ci	722	723	99.86
r =	450	453	99.34
, 95	275	277	99.28
; 95	258	260	99.23
0.05)	122	123	99.19
% vs	122	123	99.19
p =	1458	1470	99.18
; p	683	689	99.13
0.02)	104	105	99.05
0.01)	508	513	99.03

cy (WUF) and word bigram frequency (WBF), are designed, which are similar to the TF-IDF method frequently used for information retrieval [25]. In our work, the importance of a word unigram or bigram is defined as the ratio of its frequency in the result section over its frequency in the other sections.

The frequency ratios of word unigrams or bigrams are calculated based on the training set. For example, in the training set, the unigram "significantly" appears 3,514 times in the result section, but only 36 times in the other sections. Thus, its frequency ratio is 98.99% (3,514 over 3,550). All word unigrams or bigrams with frequency ratios higher than 80% were collected and examined manually by our in-lab biologists. In the Section 4, we will describe the method used to select the threshold in detail. Tables 3 and 4 show the selected top ten important word unigrams and bigrams. With the threshold 80%, we selected fifty-six word unigrams and eighty-seven word bigrams for the word unigram (WU) and word bigram (WB) lists respectively.

Based on the WU and WB lists, the f_{WUF} and f_{WBF} features are defined as follows:

$$f_{WUF}(\{WU_s\}, \{WU_{List}\}) = \begin{cases} 1, \{WU_s\} \cap \{WU_{List}\} \neq \phi \\ 0, \{WU_s\} \cap \{WU_{List}\} = \phi \end{cases},$$

$$f_{WBF}(\{WB_s\}, \{WB_{List}\}) = \begin{cases} 1, \{WB_s\} \cap \{WB_{List}\} \neq \phi \\ 0, \{WB_s\} \cap \{WB_{List}\} = \phi \end{cases},$$

where $\{WU_s\}$ or $\{WB_s\}$ indicates, respectively, the set of WUs or WBs in the current sentence; and $\{WU_{List}\}$ or $\{WB_{List}\}$ is the set of WUs or WBs in our WU and WB lists, respectively.

Take the sentence "**Twenty percent** of **pre-menopausal women** had **angiographic CAD versus 31%** of **postmenopausal women** (p = **0.02**)" as an example. First, stop words are filtered out; seventeen word unigrams are preserved shown in bold font. Then, we calculate the WUF and WBF. If the WU list contains one of the seventeen word unigrams, the feature value will be set to 1; and 0 otherwise. The same approach also applied method on the feature.

4. Results

4.1. The results section identification corpus

Since there are not any publicly available result section identification corpora, we constructed a corpus by the following procedures. First, we compiled a dictionary with dozens of obvious section tags proposed by Hirohata et al. [9]. Secondly, we use keywords, "hypertension" and "hypertension and (gene or DNA or

RNA)", to collect approximately six thousand abstracts from PubMed search service. Thirdly, we automatically remove obvious tags from the collected abstracts. Finally, our in-lab biologists check the result section boundary. Based on the corpus, we randomly selected two-thirds of the abstracts (3,808) as the training set and used remainder (1,903) as the test set to evaluate the performance of our system.

4.2. Experiment measurement

Three measurements, precision, recall, and F-measure, are used to evaluate the performance of our result identification system. The first measurement, precision, is basically to check whether how many predicting answers from CRF output that are matched with correct answers and number of predicting answers as denominator. It is defined as follow:

$$\text{Precision} = \frac{\text{Prediction Result} \cap \text{True Answers}}{\text{Predicting Result}}$$

The second measurement is recall is to check whether how many predicting answers are matched with correct answers and number of correct answers as denominator. The formula is in the following.

$$\text{Recall} = \frac{\text{Prediction Result} \cap \text{True Answers}}{\text{True Answers}}$$

But we need a formula to combine above formulas because each of those cannot individually ensure effectiveness. For example, recall is higher (95%), but if number of predicting answers are more enormous than number of correct answers, It indicates predicting answers all almost cover correct answers, which results in low precision. Therefore, final measurement called F-Measure is a mixed quantitative value is defined as follows:

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$$

4.3. Experiment design and results

We designed nine configurations to evaluate the effectiveness of each proposed feature and the following feature combinations: Position + NE, Position + NE + Tense, Position + NE + Tense + WUF, and Position + NE + Tense + WUF + WBF. Table 5 shows the evaluation results.

First, we consider each feature individually. As you can see in Table 5, the "NE" configuration yielded the lowest recall and F-measure, but the highest precision. We illustrate the reason as follows. When any NE co-occurs in both the title and the target sentence s, s is very likely to be classified as part of the results section. Therefore, the NE configuration achieves the highest precision. However, only about 60% of all sentences in the results section contain at least one NE in common with the title; therefore, the NE configuration achieves the lowest recall. The Position and Tense feature both yielded F-measure of approximately 90%, individually. Using only the WUF feature yields the best F-measure, while the WBF feature yields a comparative F-measure and the best precision.

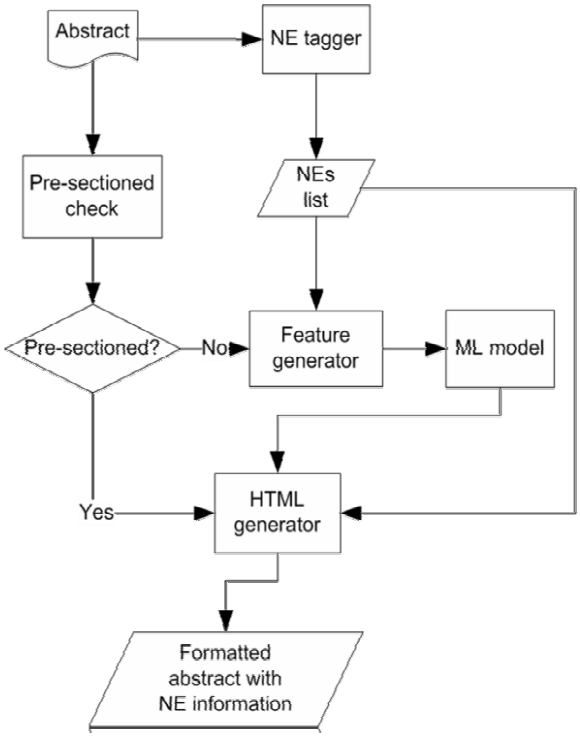

Fig. 2. The system flowchart of section identifier.

For the feature combinations, the result of the Position + NE feature shows that the NE feature can improve the precision about 6.08% against with the Position feature alone. The combination of the Position, NE and Tense features shows that, without reducing the precision too much, the Tense feature can improve the recall by 5.28% over that of Position + NE only.

In our experiment, the best result was obtained by the combination of the four proposed feature sets, Position + NE + Tense + WUF + WBF. It achieved an F-measure of 97.08% with Precision of 96.63% and Recall of 97.53%. The results show that our proposed feature sets can effectively identify sentences belonging to the result section of an abstract.

Table 5
The evaluation results: P denotes Precision, R denotes Recall and F denotes F-measure

Feature set	P (%)	R (%)	F (%)
Position	87.08	93.06	89.97
NE	97.85	60.42	74.71
Tense	90.94	89.72	89.30
WUF	94.02	96.31	95.15
WBF	95.31	90.87	93.30
Position + NE	93.16	87.01	89.98
Position + NE + Tense	92.87	92.29	92.58
Position + NE + Tense + WUF	95.72	97.18	96.45
Position + NE + Tense + WUF + WBF	96.63	97.53	97.08

4.4. A Online demo service

We have implemented an online service to demonstrate the proposed result identification approach. Figure 2 shows the flowchart of the section identifier.

For a given abstract, if the pre-sectioned check finds that the abstract contains obvious section tags, such as "Objective", and "Conclusion", the abstract is immediately divided into paragraphs. The pre-sectioned check uses a list of tag keywords collected by Hirohata et al. [9] to determine whether the abstract is pre-divided. The list keeps increasing every time our biologists or users submit new tags. If the check cannot find any obvious tags, a ML model is employed to section the given abstract.

For the section categorization problem, we first use the CRFs-based NE tagger as described in Section 3.3.2 to recognize NEs. Then a normalization module maps the found gene names to their corresponding Entrez Gene database identifiers using heuristic normalization rules [14,18]. The module relies on a lexicon containing genes and corresponding database identifiers collected from HGNC and Entrez Gene. We have also applied expansion rules to enhance the coverage of the lexicon – for example, FKBP54 is expanded to "FKBP 54" and "FKBP-54". After the normalization process is completed, successfully normalized gene names are collected into a list, and the abstract is then rechecked for NEs on the list in case the NE tagger missed any instances of the found gene names. The list is then sent to the feature generator to generate NE features.

For the section categorization problem, we applied aforementioned approach to identify the result section; we regard each sentence in an abstract as a token. Each token is associated with a boundary tag, that is the beginning (B), inside (I) or outside (O) of a section, as well as the result section tag, RS, that indicates the result section. Finally, the annotated results are sent to the HTML generator to generate formatted output.

Figures 3 and 4 show examples of section-categorized and uncategorized biomedical abstracts returned by PubMed search. As one can see, even in the categorized abstract, it is difficult to quickly differentiate the results and conclusion sections because the monochromatic text is squeezed together into one long paragraph. Setting each section apart from the others and giving it a bold heading can help researchers to focus immediately on the section of interest. Figure 5 show the same search results marked by our service. As one can see, setting the result section apart from the others and giving it a bold heading can help researchers to focus immediately on the section of interest.

5. Discussion

Several works [8,23,29,43] have showed that the Position feature is profitable for the result identification task. However, in some cases, the Position feature alone cannot correctly disambiguate the result section. For example, in the training phase, given two abstracts with the same numbers of sentences in total, but the beginning boundary of their result sections are different; one starts from the sixth sentence and the other starts from the forth. In this case, the trained CRFs model will bias on the most frequent collocation of sentence tags and positions observed in the training data. In this section, we firstly discuss the effects of combining the position feature with the proposed three feature sets. Secondly, we explain our approach for threshold selection. Finally, we explain why we chose CRFs as our ML model.

5.1. Feature combination

5.1.1. Combined with the named entity feature

In our test set, the abstract (PMID: 18269635) with the title, "*The role of* $IGF-I$ *and its binding pro-*

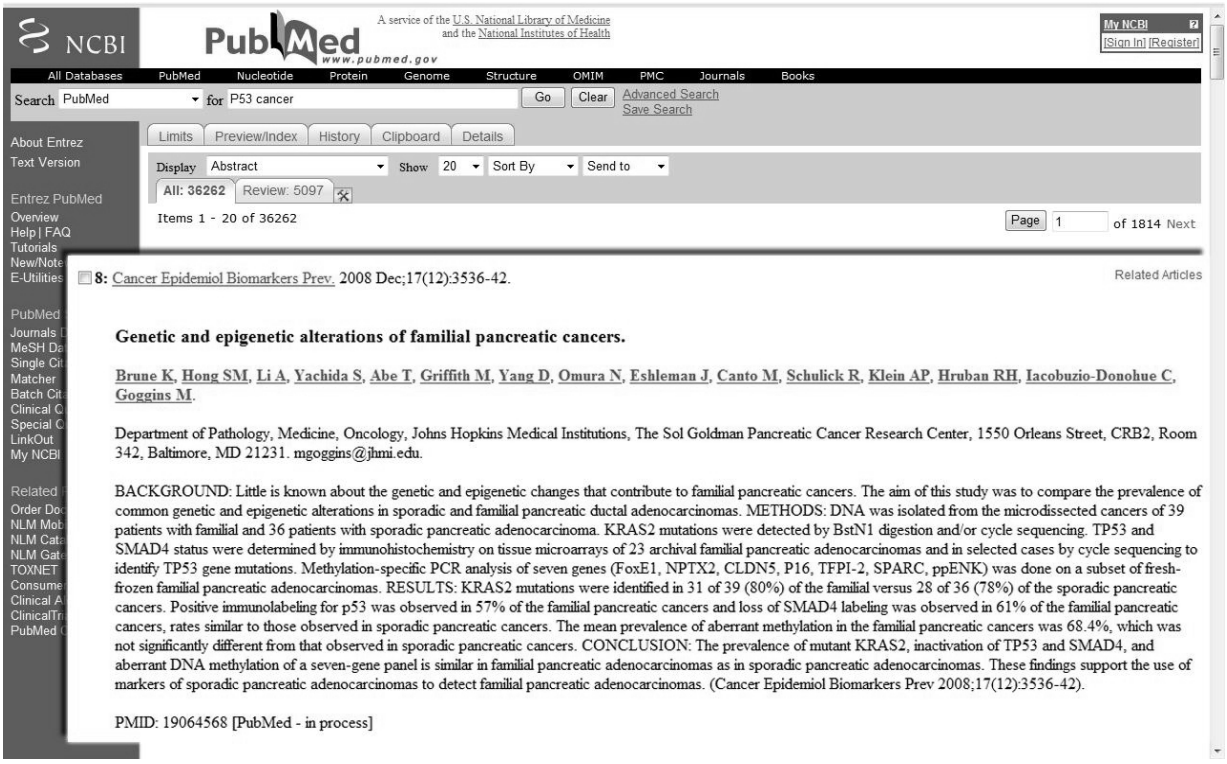

Fig. 3. The pre-categorized abstract returned by PubMed.

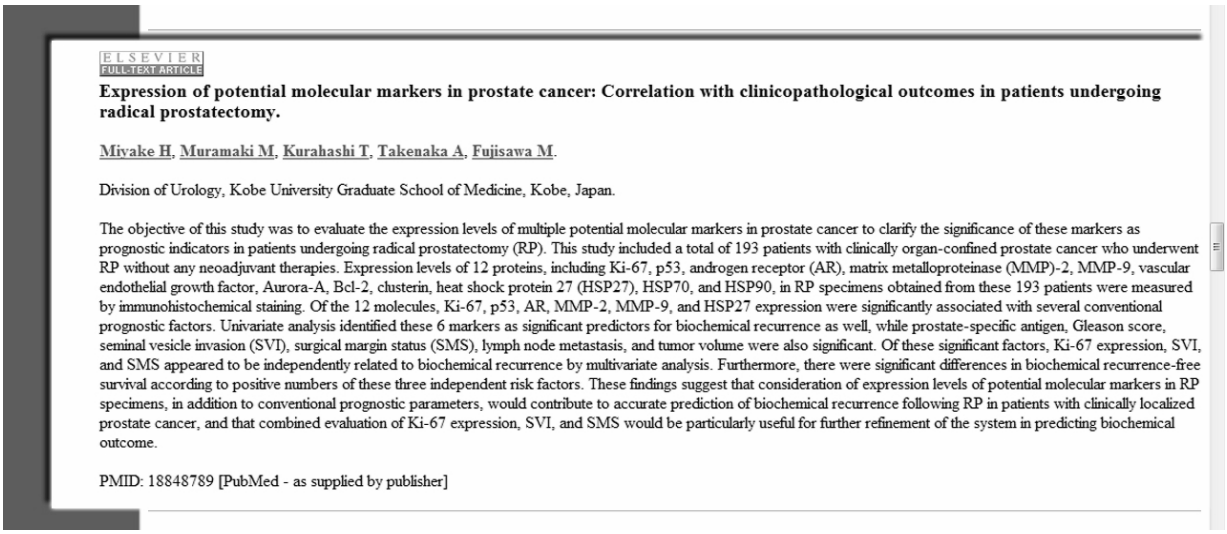

Fig. 4. The uncategorized abstract returned by PubMed.

teins in the development of type 2 diabetes and cardiovascular disease", has six sentences. The last two sentences of the abstract are annotated with the result section tag. However, the fifth sentence, "*. . . , we have focused on the potential vasculoprotective effects of both* $IGF-I$ *and IGFBP-1.*", is identified as the other section when only using the position feature. The reason why the sixth sentence misclassified is: six sentences are shared by the result section and the other sections. On average, one or two sentences may belong

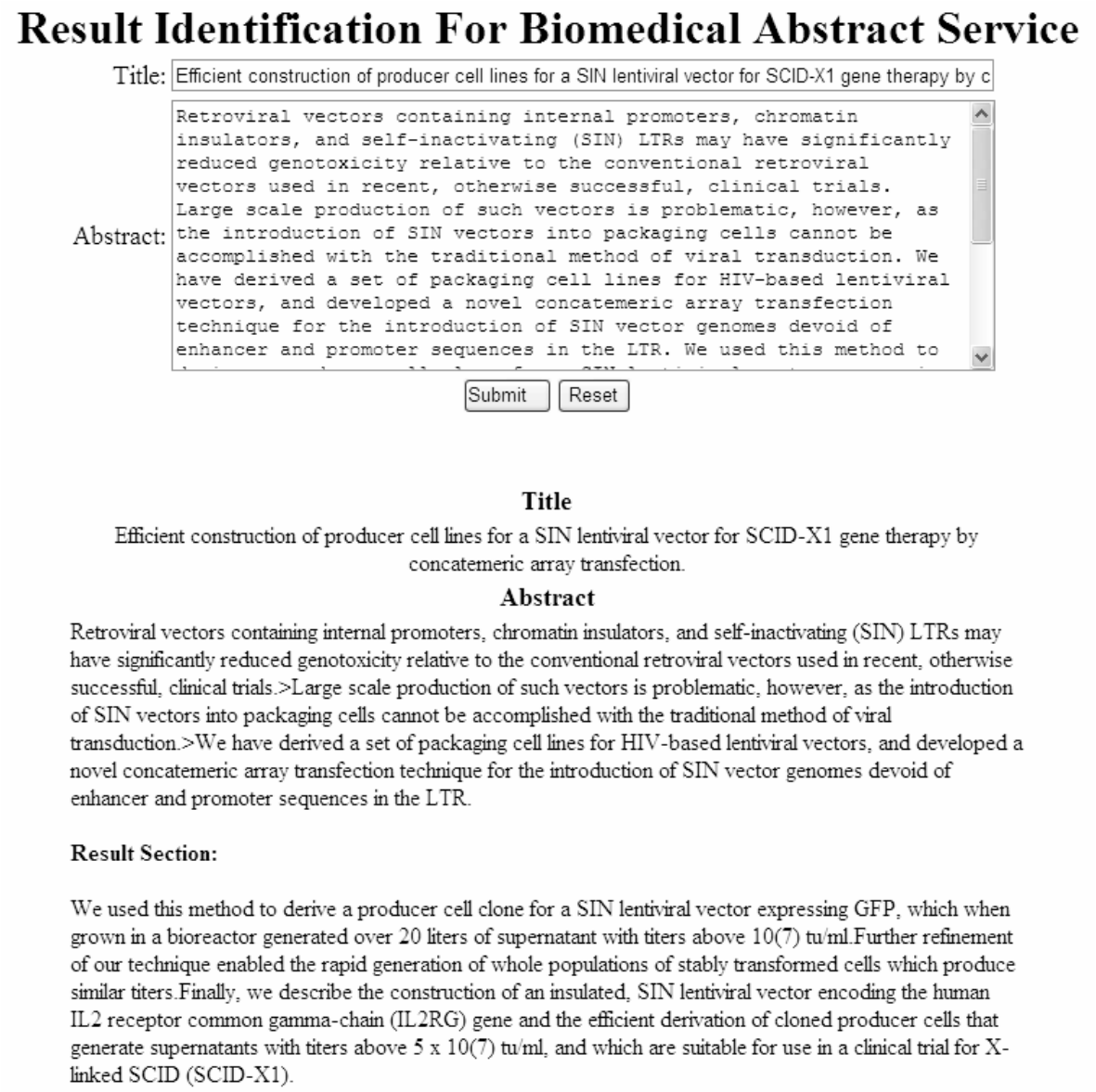

Fig. 5. The results marked up by our system.

to a section. Hence, in this case, for the Position feature only, the CRFs model cannot disambiguate well.

After NER, IGF-I is recognized as a NE, which matches the NEs in the title of the abstract. Therefore the NE feature is enabled and provides more information to our CRFs model. It guides the CRFs model to correctly identify the fifth sentence as a part of the result section.

5.1.2. Combined position + NE with the tense feature

In the beginning, we examine what the problem is in the Position + NE combination. Take the biomedical abstract (PubMed ID "16701011") for example. The abstract is comprised of fourteen sentences. The sentences of the result section are from eight to fourteen. With the Position + NE feature, sentences from six to fourteen are identified as the result section. The sixth and seventh sentences are misidentified. After analyzing sentences of the abstract, we found that the NE feature for both sentence is disabled since they don't contain any NEs. Therefore, the CRFs model only depends on the Position feature alone, which has the bias problem as aforementioned.

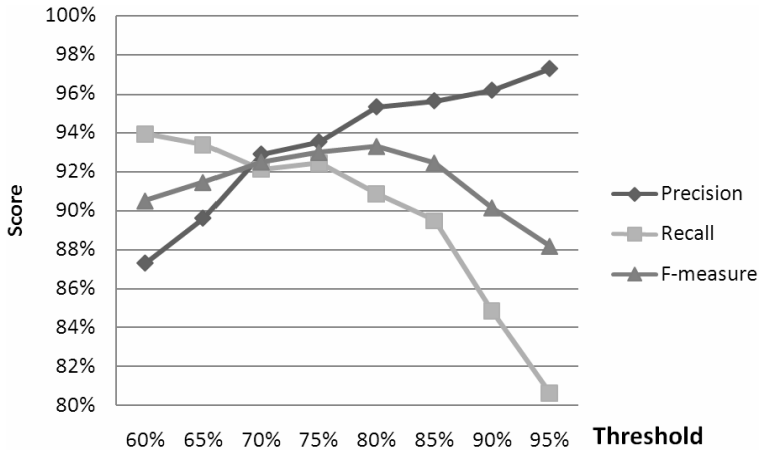

Fig. 6. Selecting word unigrams.

When combined with the Tense feature, the above problem can be solved as follows. The sixth and the seventh sentence mismatches the proposed tense of the result section as described in Section 3.3.3. In addition, although the Tense feature of the twelfth sentence does not belong to that of the result section, this sentence is still classified correctly because the sentence belongs to the tail of the abstract; competing with the Position feature, the Position feature has higher weight than the Tense feature in this case. Therefore, after introducing the Tense feature, the result section is correctly identified.

5.1.3. *Combined position + NE + tense with the word frequency feature set*

Take the biomedical abstract (PubMed ID "11593574") as an instance. The abstract consists of eleven sentences. The result section starts from five to eleven. When using Position + NE feature, the fifth sentence is misclassified as the other sections. The reasons why the sentence is misclassified include: 1) the fifth sentence locates in the middle of the abstract. Hence, the Position feature regards it as non-result section, 2) the NE feature cannot work on the fifth sentence since the sentence does not contains any NEs, and 3) the tense feature of the fifth sentence is "present perfect tense" which does not match with the tense of the result section.

Therefore, we introduce the WF feature set. The fifth sentence, "*The significant association between EH and D allele of ACE gene was found* ($P < 0.05$)", contains one word unigram, "0.05" and three word bigrams including "$p <$", "0.05)", and "(p". In this case, the WUF and WBF are both enabling. Thus, the fifth sentence eventually is classified correctly.

5.2. *Threshold selections for the word frequency feature set*

A suitable threshold to cut down the word unigram and bigram lists generated by our methods is important. In order to determine the thresholds, we randomly divided two-thirds of the training dataset described in Section 3.1 as the training set, and used the remainder as the development set. And then two experiments are conducted on the datasets, in which we gradually increased the threshold from 60% to 95%, and examined the corresponding F-measure. The thresholds with the highest F-measure are chosen. Figures 6 and 7 show the results.

As you can see a threshold of 80% yields the highest F-measure for selecting the word unigrams and bigrams. The results also show that the lower the threshold is, the higher the recall will be. In contrast, for the precision metric, a lower threshold results in lower precision. The results are in accordance with our intuition.

5.3. *Comparing CRFs-based approach with the other ML-based approaches*

In this section, we explain why we choose the CRFs as our ML model. One way to solve the section identification problem is to use the classifier-based approaches, such as Support Vector Machines [12] or Maximum Entropy [4]. In these approaches, we can take each sentence as a tag class. For result section identification, we can use the training data to train a binary classifier and apply it to determine whether each sentence belongs to result section. However, there are some problems in

Table 6
The evaluation results: P denotes Precision, R denotes Recall and F denotes F-measure

Feature set	P (%)	R (%)	F (%)
Position	91.19	89.91	90.55
Position + NE	91.19	89.91	90.55
Position + NE + Tense	91.61	92.04	91.82
Position + NE + Tense + WUF	91.29	91.21	91.25
Position + NE + Tense + WUF + WBF	93.52	94.73	94.12

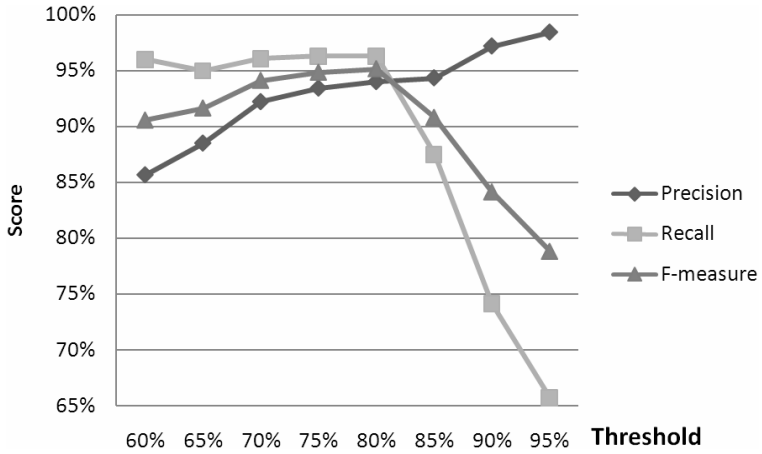

Fig. 7. Selecting word bigrams.

this approach. The most obvious flaw is that using a binary classifier to process all the sentences in an abstract causes the identified result to become segmented into many pieces. Therefore, we need a classifier that can assign a class to each sentence in sequence.

The left-to-right classifier may resolve this problem. When classifying each sentence we can rely on features from the current sentence, and the output of the classifier from previous sentence. While this technique seems to solve the problem, it makes a hard decision about each sentence before moving on to the next sentence. Hence, the classifier is unable to use information from subsequent sentences.

The Maximum Entropy Markov Model (MEMM) [17] is an augmentation of the basic ME model that incorporates the Viterbi algorithm into ME. MEMM addresses the problem of Hidden Markov Models (HMM) [45] in that HMM lies in data sparseness problem and it inappropriately uses a generative joint model to solve a conditional problem as described by Tsai et al. [38]. However, MEMM still has a label bias problem: the Markov assumptions make the transitions of MEMM leaving a given state compete only against each other, rather than against all other transitions in the model [13, 38]. Therefore, we use the CRFs model introduced by Lafferty et al. [13] to avoid the label bias problem and propose the formulation describing in Section 3.1 to transform the section identification problem into a sequential tagging problem which can be solved by CRFs.

For comparing other classifier with CRF used in this work, we utilized the same corpus described in Section 3.1 to made five experiments (Position, Position + NE, Position + NE + Tense, Position + NE + Tense + WUF, and Position + NE + Tense + WUF + WBF) using SVM. The proposed feature sets with SVM achieve F-measure, precision, and recall of 94.12%, 93.52% and 94.73%, respectively. Hence, we can see that the performance of CRF is better than that of SVM in F-measure (2.96%), precision (3.11%), and recall (2.8%). The detail of the five experiments is depicted in Table 6.

Based on these analysis, we choose the CRFs model proposed by Lafferty et al. [13] to tackle the section classification task which can avoid the label bias problem. Accordingly, we propose the formulation described in Section 3.1 to transform the section identification problem into a sequential tagging problem [2] that can be solved by the CRFs model.

5.4. The effectiveness of new feature sets

In order to clarify the effectiveness of additional feature sets, we have provided the numbers of true posi-

tive (TP), false positive (FP), and false negative (FN) cases in each configuration as follows. According to our analysis, the new errors introduced by adding a new feature come from instances in which the newly feature is disabled. Take the sentence "Arterial pressure did not differ between treated and non-treated animals" for example. It is originally correctly classified while after introducing the NE feature, it is misclassified. This is because there is no NE in it and therefore, its NE feature is disabled.

6. Conclusion

In this paper, we proposed to use the CRFs machine learning model with four feature sets to deal with the result identification problem. Several experiments are conducted to analyze the characteristics of the individual and the combination of the proposed feature sets. Our experiment results show that 1) the NE information of the title can be interpreted as features to improve the precision, 2) carefully selected word unigrams and bigrams can be a useful information to enhance both the precision and recall, and 3) the tense of a sentence can be encoded as features to improve the overall performance.

In the future work, we plan to (1) find out stable feature sets which are general enough for cross domain abstracts (not just biomedical fields), and (2) improve our approach to recognize all sections in a given abstract including the "objective" and "methods" section.

Acknowledgments

This research was supported in part by the National Science Council under grant NSC96-2752-E-001-001-PAE and NSC97-2218-E-155-001 as well as the thematic program of Academia Sinica under grant AS95ASIA02.

References

[1] P. Besnard, M.O. Cordier and Y. Moinard, Ontology-based inference for causal explanation, *Integrated Computer-Aided Engineering* **15**(4) (2008), 351–367.

[2] M. Bundschus, M. Dejori, M. Stetter, V. Tresp and H.P. Kriegel, Extraction of semantic biomedical relations from text using conditional random fields, *BMC Bioinformatics* **9**(1) (2008), 207.

[3] Q. Chen, S. Zhang and Y.-P. P. Chen, Rule-based dependency models for security protocol analysis, *Integrated Computer-Aided Engineering* **15**(4) (2008), 369–380.

[4] H. L. Chieu and H. T. Ng, A maximum entropy approach to information extraction from semi-structured and free text, *Proceedings of the Eighteenth National Conference on Artificial Intelligence* (2002), 786–791.

[5] H.-J. Dai, H.-C. Hung, R.T.-H. Tsai and W.-L. Hsu, *IASL Systems in the Gene Mention Tagging Task and Protein Interaction Article Sub-task*, in Proceedings of Second BioCreAtIvE Challenge Evaluation Workshop, 2007, 69–76.

[6] K. Fukuda, A. Tamura, T. Tsunoda and T. Takagi, Toward information extraction: identifying protein names from biological papers, *Pacific Symposium on Biocomputing* (1998), 707–718.

[7] X. Gao, L.P.B. Vuong and M. Zhang, Detecting data records in semi-structured web sites based on text token clustering, *Integrated Computer-Aided Engineering* **15**(4) (2008), 297–311.

[8] K. Hirohata, N. Okazaki, S. Ananiadou and M. Ishizuka, Identifying Sections in Scientific Abstracts using Conditional Random Fields, in: *The 3rd international Joint Conference on Natural Language Processing* Hyderabad, India, 2008.

[9] K. Hirohata, N. Okazaki, S. Ananiadou and M. Ishizuka, Identifying Sections in Scientific Abstracts using Conditional Random Fields, in: *Proceedings of 3rd International Joint Conference of Natural Language Processing* (*IJCNLP2008*) Hyderabad, India, 2008.

[10] A.N.S. Institute, American National Standard for Writing Abstracts (ANSI Z39.14-1979), 1979.

[11] K. Jin-Dong, O. Tomoko, Y.T. Yoshimasa Tsuruoka and N. Collier, Introduction to the bio-entity recognition task at JNLPBA, *Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications* (*JNLPBA-04*) (2004), 70–75.

[12] J. Kazama, T. Makino, Y. Ohta and J. Tsujii, *Tuning Support Vector Machines for Biomedical Named Entity Recognition*, in ACL-02 Workshop on Natural Language Processing in Biomedical Applications, 2002.

[13] J. Lafferty, A. McCallum and F. Pereira, *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*, in ICML-01, 2001, 282–289.

[14] W. Lau, C. Johnson and K. Becker, Rule-based Human Gene Normalization in Biomedical Text with Confidence Estimation, *Sixth Annual Computational Systems Bioinformatics Conference* **6** (2007), 371–379.

[15] J. Lin, D. Karakos, D. Demner-Fushman and S. Khudanpur, Generative Content Models for Structural Analysis of Medical Abstracts, *Proceedings of the BioNLP Workshop on Linking Natural Language Processing and Biology at HLT-NAACL* **6** (2006), 65–72.

[16] D. Marcu and A. Echihabi, *An unsupervised approach to recognizing discourse relations*, in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, 2001, 368–375.

[17] A. McCallum, D. Freitag and F. Pereira, *Maximum entropy Markov models for information extraction and segmentation*, in ICML' 00, 2000.

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

[19] C. Orasan, *Patterns in Scientific Abstracts*, 2001, 433–445.

[20] M.S. Pera and Y.K. Ng, Utilizing phrase-similarity measures for detecting and clustering informative RSS news articles,

Integrated Computer-Aided Engineering **15**(4) (2008), 331–350.

[21] S. Pyysalo, F. Ginter, J. Heimonen, J. Bjorne, J. Boberg, J. Jarvinen and T. Salakoski, BioInfer: a corpus for information extraction in the biomedical domain, *BMC Bioinformatics* **8** (2007), 50.

[22] L. Rabiner and I. A.W.A.K.-F. Lee, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, in: *Readings in Speech Recognition*, A. Weibel and K.-F. Lee, eds, 1990, pp. 267–296.

[23] P. Ruch, C. Boyer, C. Chichester, I. Tbahriti, A. Geissbuhler, P. Fabry, J. Gobeill, V. Pillet, D. Rebholz-Schuhmann and C. Lovis, Using argumentation to extract key sentences from biomedical abstracts, *International Journal of Medical Informatics* **76**(2–3) (2007), 195–200.

[24] F. Salager-Meyer, Discourse Movements in Medical English Abstracts and their linguistic exponents: A genre analysis study, *INTERFACE: Journal of Applied Linguistics* **4**(2) (1990), 107–124.

[25] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval: McGraw-Hill, Inc. New York, NY, USA, 1986.

[26] E.F.T.K. Sang and J. Veenstra, Representing text chunks, in EACL-99, 1999.

[27] B. Settles, *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), 2004.

[28] F. Sha and F. Pereira, Shallow parsing with conditional random fields, *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* **1** (2003), 134–141.

[29] M. Shimbo, T. Yamasaki and Y. Matsumoto, Using sectioning information for text retrieval: a case study with the MEDLINE abstracts, in: *Proceedings of Second International Workshop on Active Mining* (*AM'03*), 2003.

[30] L. Smith, L. Tanabe, R. Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. Friedrich, K. Ganchev, M. Torii, H. Liu, B. Haddow, C. Struble, R. Povinelli, A. Vlachos, W. Baumgartner, L. Hunter, B. Carpenter, R. Tsai, H.-J. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. Adriaans, C. Blaschke, R. Torres, M. Neves, P. Nakov, A. Divoli, M. Mana-Lopez, J. Mata and W.J. Wilbur, Overview of BioCreative II gene mention recognition, *Genome Biology* **9**(Suppl 2) (2008), S2.

[31] T.F. Smith and M.S. Waterman, Identification Of Common Molecular Subsequences, *Journal of Molecular Biology* **147** (1981), 195–197.

[32] C. Sporleder and A. Lascarides, Exploiting linguistic cues to classify rhetorical relations, *Recent Advances in Natural Language Processing IV: Selected Papers from RANLP 2005* (2007), 157.

[33] C. Sutton and A. McCallum, An Introduction to Conditional Random Fields for Relational Learning, *Introduction to statistical relational learning* (2007), 93.

[34] J.M. Swales, Genre analysis: English in academic and research settings, 1990.

[35] X. Tao, Y. Li and R. Nayak, A knowledge retrieval model using ontology mining and user profiling, *Integrated Computer-Aided Engineering* **15**(4) (2008), 313–329.

[36] S. Teufel and M. Moens, Summarizing scientific articles: experiments with relevance and rhetorical status, *Computational Linguistics* **28**(4) (2002), 409–445.

[37] R.T.-H. Tsai, C.-L. Sung, H.-J. Dai, H.-C. Hung, T.-Y. Sung and W.-L. Hsu, NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition, *BMC Bioinformatics* **7**(Suppl 5) (2006), S11.

[38] R.T.-H. Tsai, C.-L. Sung, H.-J. Dai, H.-C. Hung, T.-Y. Sung and W.-L. Hsu, NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition, *BMC Bioinformatics* **7**(Suppl 5) (2006), S11.

[39] R.T.H. Tsai, C.L. Sung, H.J. Dai, H.C. Hung, T.Y. Sung and W.L. Hsu, NERBio: using selected word conjunctions, term normalization, and global patterns to improve biomedical named entity recognition, *BMC Bioinformatics*, 2007.

[40] Y. Tsuruoka, Y. Tateishi, J. Kim, T. Ohta, J. McNaught, S. Ananiadou and J. Tsujii, Developing a Robust Part-of-Speech Tagger for Biomedical Text, *Lecture Notes in Computer Science* **3746** (2005), 382.

[41] R. Weissberg and S. Buker, Writing up research: experimental research report writing for students of English: Prentice Hall Regents, 1990.

[42] J.C. Wu, Y.C. Chang, H.C. Liou and J.S. Chang, Computational analysis of move structures in academic abstracts, *Proceedings of the COLING/ACL on Interactive Presentation Sessions* (2006), 41–44.

[43] Y. Yamamoto and T. Takagi, A Sentence Classification System for Multi Biomedical Literature Summarization, *Proceedings of the 21st International Conference on Data Engineering*, 2005.

[44] Y. Yusof and O.F. Rana, Combining structure and function-based descriptors for component retrieval in software digital libraries, *Integrated Computer-Aided Engineering* **15**(4) (2008), 279–296.

[45] S. Zhao, *Named Entity Recognition in Biomedical Texts using an HMM Model*, in COLING 2004 International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA), 2004.