Improved Named Entity Recognition - Patterns in Columns Model (PCM)

Chun Yong Moon,Manjula Pilaka,Hye-Young Paik, John Shepherd
School of Comp Sci and Eng, UNSW, Sydney, Australia
mpilaka,hpaik, jas@cse.unsw.edu.au

ABSTRACT

The goal of this paper is to improve the Named Entity Recognition for automatic information extraction related to record based data in text documents. This paper used Patterns in Columns Model by identifying patterns in record based data using delimiters and tags. Our approach has been implemented and evaluated against call for paper documents. Our model showed the best recognition rate and significant results compared to existing NERs.

Keywords

PCM - Patterns in Columns Model, DBA - Delimiter based patterns in columns approach, TBA - Tag based patterns in columns approach, HMM - Hidden Markov Model, CFP - Call-for-papers

1. INTRODUCTION

The quantity of information available online is growing at an ever-increasing rate and the sheer volume of data makes finding useful information difficult, despite the heroic efforts of search engines like Google and Bing. Search engines, however, are inherently limited since they are based on keyword-driven search over a massive body of documents (web pages). Approaches such as the Information Extraction (IE) aims to extract structured data out of the vast collection of unstructured or semi-structured data that comprises the Web. One of the main concerns in IE is to recognise and extract core Named Entities and their relationships in given documents. Using the entities and relationships, for example, one can issue a more fined grained search/query over the documents.In the last decade, the IE research community has produced a number of Named Entity Recognizers (NER) that are now widely used. However, the tools are generally designed for sentence-based text. To our best knowledge, the possibility of applying the tools to record-based text has not been fully explored.

SAC'14 March 24-28, 2014, Gyeongju, Korea.

This study is aimed at improving the performance of named entity recognition task for IE in record-based data. The core principles in designing our approach are that (i) we use the available IE tools 'as-is' without domain-specific training or customisation, (ii) we focus on exploiting the inherent record structure in source data.
In this paper, we introduce our **Patterns in Columns** model (PCM). It proposes to achieve effective NER by:

- identifying a region of the text containing relevant record-based data,
- using existing NER systems to classify entities in this region,
- using the (partially-correct) classification to identify record structures,
- using the record structures to correct errors in the original NER.

Using feedback between entity recognition and record-based structure identification, we are able to achieve significant improvements in the initial NER phase.

The rest of this paper is organised as follows: In Section 2, we use call-for-papers scenario from the acadamic conferences program committees data as record based data for our experiments to improve NER entity recognition. In Section 3,we detail the previous NER systems used to retrieve record based data. In Section 4, we discuss the Patterns in Columns model and its components in detail.In Section 5, we detail the effectiveness and efficiency of the retrieval of record based data using Patterns in Columns model as compared to the previous NER systems.

2. APPLICATION SCENARIO

Information extraction does not generally aim to extract all entities and all relationships from every document, but typically works with respect to a target schema. The schema describes what kind of entities and relationships are relevant to an application and the IE system then builds a database to support the application by finding examples of such entities and relationships in a document collection and using this to populate the database.
In this paper, the target schema is a database of information about researchers, their work (such as publications and projects), the organizations they work for, Research-related events (such as conferences and seminars), and so on. We focus on one particular relationship (people and their affiliations with organizations).


CALL FOR PAPERS
The 17th International Conference on
Data Engineering=20
April 2-6, 2001
Heidelberg, Germany
Sponsored by the IEEE Computer Society
ICDE 2001 in the Heart of Europe

The 2001 International Conference on Data Engineering (ICDE 2001) provides
a significant opportunity to:
1. Attend one of the worlds premier forums that deals with the use of ~~~~~~~~~~~~~~~~~~

Important Dates:
January 15, 2001 : Submission deadline for Tutorial Proposals
February 15, 2001 : Decision on Tutorial Program.

For more information in TUTORIALS and/or SPECIAL SESSIONS can be found
at the website: http://icip01.ics.forth.gr.

PROGRAM COMMITTEE
Chair: Michael Gertz — University of Heidelberg, Germany, www.mglab.com
Co-Chair: Johann-Freytag — Humboldt-Universitaet zu Berlin, Germany, www.jff.edu.gy
Carlo Batini — University of Milano - Bicocca, Italy, www.um.edu.it

Figure 1: Record-based data (indicated by surrounding boxes) in CFPs

Calls-for-Papers (CFPs) for academic conferences are a useful class of documents, since they provide a rich source of data about researchers and their affiliations. CFPs are widely available; they are frequently distributed by email (push) but are also accessible on the web (pull). For the purpose of demonstrating the PCM model, we used CFP documents as our primary data source. Note that while this work has focused on CFP documents and a specific target schema, the methods are applicable to any record-based data and compatible target schema. An important observation on CFPs is that they are not simple prose, but generally use some informal structuring conventions. In particular, record based data, appears reasonably frequently. Figure 1 shows such examples. When the writer of a CFP makes a document, he/she gives it a title, divides it into sections, and uses layout in order to deliver his/her intention precisely to readers. One frequently-used informal structuring mechanism is lists-of-records. These are often realized via column layout or by using list markers and delimiters to separate fields. Such structures clearly carry additional information beyond the written words.
Extracting conference data into structured form provides for a range of potential applications. For example, a precision conference search service, which allows conferences to be discovered based on queries like "Computer Science related conferences in Sydney during November of 2014". As another example, a subscription service might let users register their interest in certain kinds of conferences and be notified when such conferences are added to the database. Both of these applications assume that a stream of CFPs can be supplied to the system (e.g. via subscription to multiple mailing lists) in order to maintain the database.

The specific problem that we target in this paper is the accurate identification of people, organizations and locations within lists of committee members in CFPs. Such lists are typically presented as a sequence of records, where each record gives information about one committee member; the person's name is always present, but is generally accompanied either by an organization, a country, or both.

Records may be arranged either in tabular form, using columns to indicate field locations, or as lists, using a variety of delimiters to separate records. Consider the program committee membership list. Ideally, we would recognize this as a sequence of records, separated by hyphens, with each record containing the name of a person and the name of an organization, separated by a comma. We would extract core conference information, and build a record for the conference itself, and also associate people with their roles in the conference (e.g. General Chair, PC member...).

3. RELATED WORK

3.1 Information Extraction from CFPs

Schneider [11] describes a system that uses IE to extract information from conference announcements in order to determine whether/how they should be included in a conference directory. The study focuses on conferences in the areas of linguistics and computational linguistics.One goal is to assign each conference to a node in a topic hierarchy.The information to be extracted from each CFP (the target schema) was name, title, dates, location and url. Because the information is available as a plain-text email body, and generally does not follow english grammar rules (except in the descriptive paragraphs), Schneider did not use any POS-tagging but used lexical and orthographic properties of the text to identify the relevant pieces. In addition, he used a cascade of finite state transducers for tokenization and tagging. Schneider's system was very efficient to execute but did not produce particularly effective extraction (except for dates and URLs); the F-measure for conference location, for example,was around 75%. He identified a number of scenarios where extraction errors occurred (e.g. conference titles that do not contain a keyword like "Conference" or "Workshop" or "Symposium" were not detected), and proposed heuristic fixes for some of these. To evaluate the system, Schneider used a set of 263 CFPs, half of which was used for training and half of which was used for testing. He evaluated the system using a variety of feature sets, in order to determine which fatures were the most useful. He found that using more features produced better results, but the overall F-measure, based on how accurately individual words were tagged, was not particularly high (around 80%). Considering individual slots in the target schema, "conjoined" was the least accurate (F-measure of 51%), with "name" being next (around 58%), and the remaining features ("date", "location", etc.) around 70%. Schneider attributes the relatively poor performance to the size of the training set and to the fact that CFPs don't exhibit the same kind of grammatical structure as do the kinds of documents studied in other work (such as the research paper IE task in [5]).

3.2 Column and Pattern-Based Recognition

This section will describe the research concerning the column based and pattern based recognition. We looked into the table recognition method which use conditional random fields (CRFs) to extract the table. Pinto et al [7] used the CRF method to extract the table and he also compared the results with the methods attained from the Hidden Markov Models. They used the plain-text government statistical reports as their data.

Information in many documents is delivered not only through the stream of words but also through the layout of the words. A great example which the information is delivered through the layout of words would be the table. Extracting information from the table is one of the most difficult tasks in

QuASM system (Question Answering Using Semi-structured Metadata[6]). Question Answering System usually follows two retrieval steps.First, retrieve the appropriate documents to attain the appropriate answers. However, in this process QuASM extracts too much information from the table.

This research will describe the pattern based recognition from our research approaches. Many research patterns attempted the information extraction. Out of those, Xiaoyan Li et al. [4] have suggested a new novelty detection approach based on the identification of sentence level patterns. Their study focused on the identification. Xiaoyan Li et inspired his ideas from the question answering technique which locates the wrongly linked patterns from the sentences. They proposed to first extract interesting sentences with certain patterns that included both query words and required answer types, indicating the presence of potential answers to the questions, and then identified novel sentences that were more likely to have new answers to the questions. This study elaborated an analysis of sentence level patterns, focusing on named entities, with the data from the TREC 2002 novelty track data. and provided the proposed pattern-based approach to novelty detection[9]. Out of the researches done on patterns, there is a representative research concerning the statistical method, CRFs. Watanabe et al. [13] studied methods to differentiate named entities in Wikipedia[13, 8, 12]. He proposed ways to categorize the Conditional Random Fields (CRFs) with the nodes' graph and he also introduced how anchor texts can be recognized as graph structure through nodes. They reduced the computational cost through the inference method based on the Tree based Reparameterization. In addition, they compared their methods to the one which uses the Support Vector Machines (svm) to show its relative effectiveness. Wikipedia articles are semi-structured texts. Especially the fact that some elements have dependencies among each other is a important characteristic between list (<UL> or <OL>) and table (<TABLE>). They focused on lists that appear most frequently in Wikipedia. This research introduced the Tree-based Reparameterization (TRP) to calculate the marginal probabilities. They used 14285 NEs, 16136 anchor texts and 2,300 articles from Japanese Wikipedia as their dataset. They researched the effect of ecach cliques and compared the method recommended from CRFs and the baseline method suggested from SVMs. Their results showed that the dependency information from the HTML tree help to categorize entities without gloss texts in Wikipedia. However statistically differentiating has its down side because insufficient positive examples makes the labeling difficult.

4. PATTERNS IN COLUMNS MODEL (PCM)

Our PCM model includes three distinct modules, namely: NER File Reader, Named Entity Resolution and Interpolated Named Entity Resolution. The simple data flow between the components is shown in Figure 2. NER File Reader firstly extracts the Program Committee "region" which contains a record based type data. The extracted Program Committee sections are fed into Named Entity Resolution module which recognises named entities and tags each entity with an appropriate label (e.g., Person, Location). The tagged Program Committee sections are sent to Interpolated Named Entity Resolution module. Then, the Interpolated Named Entity Resolution module resolves any mistakes (i.e. Mislabelling, unrecognised entities). We explain each module in detail in the following.

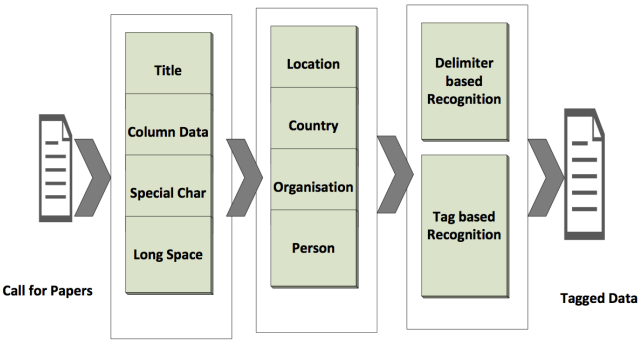

Figure 2: System Overview

4.1 NER File Reader

NER File Reader pre-processes the data in 5 steps listed below and extracts data.

Syntax and Regular Pattern Recognition: On receiving the input data, NER File Reader reads the title, extracts the record-based data following the title "Program Committee". The data is extracted as record-based data (or also referred to as column type data in this system).

Processing Special Characters: NER File Reader removes special characters such as *located at the start or the end of a sentence. Majority of the column type data often uses the special characters in front of the sentence as a new line indicator. However, the special characters can sometimes cause the named entity recognisers to behave unexpectedly, so we remove them in this step.

Processing Column Delimiters: We need to identify potential column delimiters. We treat a double space or more as a long space. Besides the long spaces, other delimiters like a comma (,) hyphen (-), or tab are also common in Column Type Data. Both LBJNER and Stanford CRF-NER [3] can perform poorly when the input has long spaces. There are many cases that the long space is failed to be recognized as the delimiter distinguishing between chunks whereas two chunks are recognized as one chunk because a long space is recognized as one space.

Processing General Patterns: We pre-process what we Call "General Patterns" such as URL and e-mail addresses For example, an URL or e-mail address can be described by the regular expression as below.

- **URL:** ^ (https?\/\/)? ([\da-z\.-]+)\. ([\a-z\.]{2, 6})([\/\w\.-]*)*\/?$
- **E-mail address:** ^ [a-zA-Z0-9\] +@[[a-zA-Z0-9\] +.][A-zA-Z0-9\]$

4.2 Named Entity Resolution

In designing of this module, we have evaluated the three chosen systems LBJ NER Trigger [10], CRF-NER [3], ANNIE [2] (due to their availability and popularity in IE community), and analysed their performance on individual class

of entities (i.e., Person, Location, Organization, etc.). We have observed that instead of picking an overall winner in recognising all entities, it is better to identify a better performer in each class and combine the results.

Hence, the core of this module is the logic of combining results from the underlying NER systems that we use. The module has strengthened its function with the result of union tests of the three systems that we have used out-of-the-box without training or customisation.

We recognize Country names first. The reason why the Country is recognized before other entities is because the names of Country have few changes or new production and also its size of the dictionary is relatively small comparing to other entities.

System uses gazetteers of LBJ NER Tagger [10] as a dictionary in order to form the Country Pattern. LBJ NER performs the best out of the three systems we evaluated on this Country entity class. However, despite the highest recognition rate, LBJ NER also has an unclassified entity rate for location and country which decreases the recognition rate of this system overall. In order to improve the performance, we also combine the results of other NERs for this entity class as well (in the Union Tagger module).

The Union Tagger with the help of API of Stanford CRF-NER has been applied to recognize the Organization. This is because the recognition rate of CRF-NER is 63%, and that of LBJNER is 25% whereas that of ANNIE is just 49% which needed to be excluded. In addition, LBJNER has also been excluded since it often recognizes the Organization as the Person which is incorrect and such errors could decrease the recognition rate of Stanford CRF-NER.

However, the recognition rate of CRF-NER is over 50% fully, so it has been a sufficient base to strengthen the recognition rate in Named Entity Resolution. Since Stanford CRF-NER has 69% of the recognition rate and LBJNER has 66% of that on recognizing the names of Person, these two have been applied by combining with Union Tagger. ANNIE has been excluded from the Union Tagger due to its 24% recognition rate.

4.3 Interpolated Named Entity Resolution

This module rectifies the errors of the previous module by utilising the record based structure of the input data. We use the following two approaches: delimiter-based and tag-based.

4.3.1 *Delimiter-based PCM approach (DBA)*

In this section, the first approach of PCM, named *Delimiter-Based PCM Approach (DBA)* is introduced. We perform an analysis to discover what is called "Orthographic Row Type Patterns". This is based on observation that record-based data has a pattern repeated in each 'row' and such a feature can be used in order to improve the entity recognition rate.

Each pattern is expressed with Tags and Delimiters. Table 5.1 shows an example of tags and delimiters used in building orthographic row type patterns; the "Represents" column gives a specific example of each tag type. Applied as in Country Tagger and Union Tagger, all the classified entities have been represented with such a Tag. Figure 3 shows an example of orthographic row type patterns.

If there is one type in one region, it does not require correction whereas if diverse types appear in one region, errors should be corrected with multi Type Row process.

PROGRAM COMMITTEE

<Unknown>: <Person>//<Organization>,<Country>,<URL>
<Unknown>: <Person>//<Organization>,<Country>,<URL>
<Person>//<Organization> - <Location>,<URL>
<Person>//<Organization>,<Country>,<URL>
<Person>//<Organization>,<Country>,<URL>
<Person>//<Organization>,<Country>,<URL>
<Person>//<Organization>,<Country>,<URL>
<Person>//<Organization>,<Country>,<URL>
<Person>//<Organization> - <Location>,<Country>,<URL>
<Person>//<Organization>,<Country>,<URL>
<Person>//<Organization>,<Country>,<URL>
<Person>//<Organization>(<Unknown>),<Country>,<URL>

Figure 3: Orthographic Row Type Pattern

Table 1: Example of Tags and Delimiters

	Notation	Represents
Tag	<Person>	Person name type
	<Organization>	Organization name type
	<Location>	Location name type
	<Country>	Country name type
	<Email>	Email address
	<URL>	URL address
	<Date>	Date type
	<Unknown>	Unclassified Text Slot
Delimiter	, { } ( ), [ ], -, _	Special Char
	...	Dots
	//	Big space between words

In the multi Type Row process, Types with high value get matched with comparison of the statistical values of the Types. This process has a similar pattern to HMM that is to find the most appropriate pattern by using each state probability and transition probability. The transition probability is to illustrate the probability of current correct answers based on previous ones, and the state probability is to illustrate correct answers about the current state with the probability. In short, HMM (Hidden Markov Model) is a method of multiplying the transition probability and state probability. In this process, calculating state probability of each chunk is followed by calculating transition probabilities of each row. From statistical test result, union effect usually performed approximately between 60% - 90%. Such produced entities are fixed by DBA.

In order to figure out pattern probability, categorising the numbers of chunk and delimiter needs to be done first. Secondly, the Transition Probability of each row needs to be figured in order to find out main pattern for each group. The idea is similar to Viterbi algorithm applied Greedy algorithm. The biggest feature of the Viterbi algorithm is to calculate by storing the highest probability of its spot among calculated probabilities in each node. As the Greedy Algorithm methods are, Optimal Solution cannot be guaranteed, but the close answer to the Optimal Solution can be determined quickly. This approach recognizes a pattern with the highest probability value among probabilities of each row as main pattern of the groups. Lastly, the system will fix unclassified entities or incorrectly classified entities by using this main pattern.

4.3.2 *Tag-based PCM approach (TBA)*

In this section, we introduced the second approach to PCM, named *Tag-based PCM approach (TBA)*. The previous approach assumes the input to have regular/repeated record structure. However, it is hard to apply the previous approach method when the record structure is different to what is assumed in the approach (e.g., no line breaks, or multiple lines in one logical record). The second approach attempts to design a more general method which can be applied to record based data that do not have the line break per record.

In this approach, the common part of the system (NER File Reader, Named Entity Resolution) remains the same. The difference is that we apply entity resolution algorithms (in Named Entity Resolution module) after transforming all into a single line data, followed by an analysis of patterns *in the line of sequence*. That is, we remove the notion of lines and treat the data as a sequential line of words.

James kim , IBM , UK , Kim J , UTS , US , Mark Park , sydney Uni. , USA , J S Lee , KAIST, Korea , K J Jang , POSTEC , Japan , Milcle Lee , U. Lundon , UK , Chatswood , James Kang , UTS , UK , Wales , Big Bug , ANU , AU , Sydney , Dong H Kim , UNSW , AU , Hills

Figure 4: Example of Input Data

We introduce an algorithm for recognizing entities by calculating a probability of various features and that of repeated patterns. Ultimately, the algorithm aims to identify columns in a record by mathematically analysing repeated patterns from the input. The approach consists of two sub modules. The first sub-module, Line Pattern Recognition module has a function to recognize and tag unknown entity after calculating each pattern features and probabilities. The second sub-module, Line Group module is to produce columns in a record from repeated pattern by grouping. Both the algorithms assume there are maximum of W+3 chunks of patterns.

Line Pattern Recognition Sub-Module.

There are unrecognised named data in Union Tagger. This data becomes an input data for this sub module. These errors can be corrected with this algorithm. The main steps in this sub module are:

1. Feature Vector Creator receives data from Named Entity Resolution as input, then analyses features of the input data.
2. A probability of the features analysed by the module is calculated and the module produces a feature vector by storing it.
3. The module begins to search matched pattern pairs using the feature vector as the basis.
4. The module tags by fitting unknown named entities into main pattern and the module tags the rest of untagged named entities after searching the patterns fitted in an order of each feature.

Line Grouping Sub-Module.

Line Grouping Module analyses repeated patterns, groups into each pattern and then transforms into a column in order to make tagged result data appear column type. The data tagged by the Line Pattern Recognition sub-module will be treated as input to the Line Grouping Sub-Module and now tagged as column type from Line type to fix the remaining unknown entities in the data. The main steps in this sub module are:

1. Decide group size.
2. After determining the group size, integrate patterns and count the frequency of the patterns.
3. Decide main pattern among pattern candidates. Count frequency of analysed patterns.
4. Change into column type after type grouping in the main pattern. The data not grouped by the logic above will be grouped by themselves. As a result, there will be no chunks left that are not grouped.

Algorithm 1 Algorithm for Line-Pattern Recognition Sub-module

Require: Union Tagger Output data as Input
  Feature data list(Probability of each feature)
  Split Chunks Column *token*
  for *token* in $1-3$ **do**
    GetPatternPair from MainPatternRecognition($Word$,$Word$+ *token*)
    if Feature Vector Creator($Word$,$Word$ + *token*) == Pattern Pair($Word$,$Word$ + *token*) **then**
      Count Frequency of Relevant Pattern Pairs
    end if
  end for
  if *token* > 3 **then**
    Get Orthogonal Pattern Features with Feature Vector Creator
  end if
  return *OutputData*

2-gram pattern	Frequency	Description
(Org, Org)	1	Exception
(Org, Cou)	7	
(Cou, Per)	4	
(Per, Org)	6	
(Cou, Loc)	3	
(Loc, Per)	2	

Algorithm 2 Algorithm for Line-Grouping Sub-module

Require: Union Tagger data without line breaks as Input
  Decide *GroupSize*
  if *PatternFrequency* > 1 **then**
    Count 2-gram Pattern Frequency
  end if
  for *GroupSize* =2 to *PatternFrequency* **do**
    IntegratePatterns (*GroupSize*)
    Count Frequency(Analysed Patterns)
  end for
  for *GroupSize* $= 4$ to 2 **do**
    Identify Main Pattern Matching Algorithm (*GroupSize*,*MainPattern*)
    Count Frequency (Analysed Patterns)
  end for
  Change to Column type
  Group Pattern data
  return *OutputData*

5. EXPERIMENTS AND RESULTS

5.1 Experiments

Two sets of experiments were performed. First, using a subset of CFPs documents from 1994 and 2010, we evaluated the three underlying systems (i.e., Annie, CRF-NER and LBJ-NER) in order to compare their performance against each other on the different types of named entities. We used the results as the basis for designing the Union Tagger as discussed in Section 4.2. The second experiment focused on evaluating PCM. We used 23 cases of Program Committees data extracted out of the call-for-paper documents in year 2010. The size of the data was 116MB overall including 2,082 entities. We measured the number of classified entities, unclassified entities, correctly classified entities, and incorrectly classified entities from the three underlying systems as well as our own solutions in PCM.

5.2 Results

The Person Entity recognition rate has been increased by 12% more by union tagger of PCM. It has been increased by 18% more by DBA of PCM. Totally, it has been increased by 30% more by PCM. The Organization Entity recognition rate has been increased by 3% more by union tagger of PCM. It has been increased by 29% more by DBA. Totally, it has been increased by 32% more by PCM. As a result, compared to the highest performance NER among three existing NERs, the Location/Country Entity recognition rate has been increased by 8% more by union tagger of PCM. It has been increased by 4% more by DBA. Totally, it has been increased by 12% more by PCM. Up to now, the first approach (DBA) showed a significant improvement in extracting record-based data in CFPs. The recognition rate

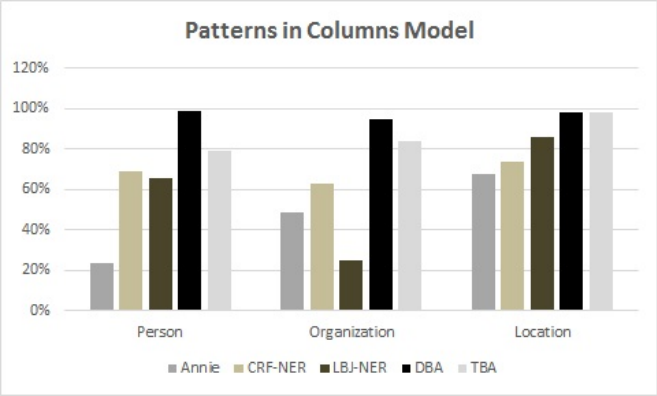

Figure 5: Results

about Person Entity has been increased by 10% more by TBA of PCM. Compared to the highest performance NER among three existing NERs, The recognition rate about Organization entity has been increased by 21% more by TBA. Compared to the highest performance NER among three existing NERs. The recognition rate about Location/Country Entity has been increased by 12% more by TBA module of PCM. More detailed setup of the experiments and itemised results can be found in [1].

6. CONCLUSION

In this paper, we proposed an IE approach where we integrate existing NER results with our error correction algorithms (namely DBA and TBA) to improve the entity recognition rate. Our approach does not require data training or domain specific customisation.

As future work, we plan to explore applying orthographic patterns more effectively in TBA and experiment with other possible sequence patterns to increase accuracy. We also plan to investigate more reliable entity instance resolution techniques by applying our current solution to a real application. For example, we plan to map our extracted data to a target relational schema and build an application to track people and their affiliation relationships with organisations.

7. REFERENCES

[1] Authors Removed For Blind Review. Improved named entity recognition for information extraction in record-based data. Masters Thesis, 2013.

[2] H. Cunningham. *Developing Language Processing Components with GATE Version 6*. The University of Sheffield, Department of Computer Science, 2010.

[3] J.R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, 2005.

[4] X. Li and B. Croft. Novelty detection based on sentence level patterns. In *CIKM*, pages 744–751. ACM, 2005.

[5] F. Peng and A. McCallum. Information extraction from research papers using conditional random fields. *Information Processing & Management*, 42(4):963–979, 2006.

[6] D. Pinto, M. Branstein, R. Coleman, et al. QuASM: a system for question answering using semi-structured data. In *2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 46–55. ACM, 2002.

[7] D. Pinto, A. McCallum, X. Wei, and B. Croft. Table extraction using conditional random fields. In *SIGIR*, pages 235–242. ACM, 2003.

[8] S. P. Ponzetto and M. Strube. Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution. In *HLT-NAACL*, pages 192–199. Association for Computational Linguistics, 2006.

[9] H. Qi, J. Otterbacher, A. Winkel, and D. R. Radev. The University of Michigan at TREC2002: Question answering and novelty tracks. Technical report, 2003.

[10] L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 2009.

[11] KM Schneider. Using information extraction to build a directory of conference announcements. In *Computational Linguistics and Intelligent Text Processing*, pages 521–532. Springer, 2004.

[12] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, pages 1419–1424, 2006.

[13] Y. Watanabe, M. Asahara, and Y. Matsumoto. A graph-based approach to named entity categorization in wikipedia using conditional random fields. In *EMNLP-CoNLL*, pages 649–657, 2007.