International Journal of

# Computational Linguistics & Chinese Language Processing

# 中文計算語言學期刊

A Publication of the Association for Computational Linguistics and Chinese Language Processing

This journal is included in THCI, Linguistics Abstracts, and ACL Anthology.

易繫辭曰上古結繩而
治後世聖人易之以書
契百官以治萬民以察
說文敘曰蓋文字者經
藝之本宣教明化之始
前人所以垂後後人所
以識古故曰本立而道
生知天下之至賾而不
可亂也教化既萌文心
雕龍則謂人之立言因
字而生句積句而成章
積章而成篇篇之彪炳

Vol.19 No.1 March 2014 ISSN: 1027-376X

# International Journal of Computational Linguistics & Chinese Language Processing

**International Journal of**

# Computational Linguistics & Chinese Language Processing

## Aims and Scope

**International Journal of Computational Linguistics and Chinese Language Processing** (IJCLCLP) is an international journal published by the Association for Computational Linguistics and Chinese Language Processing (ACLCLP). This journal was founded in August 1996 and is published four issues per year since 2005. This journal covers all aspects related to computational linguistics and speech/text processing of all natural languages. Possible topics for manuscript submitted to the journal include, but are not limited to:

- Computational Linguistics
- Natural Language Processing
- Machine Translation
- Language Generation
- Language Learning
- Speech Analysis/Synthesis
- Speech Recognition/Understanding
- Spoken Dialog Systems
- Information Retrieval and Extraction
- Web Information Extraction/Mining
- Corpus Linguistics
- Multilingual/Cross-lingual Language Processing

## Membership & Subscriptions

If you are interested in joining ACLCLP, please see appendix for further information.

## Copyright

## Cover

Calligraphy by Professor Ching-Chun Hsieh, founding president of ACLCLP
Text excerpted and compiled from ancient Chinese classics, dating back to 700 B.C.
This calligraphy honors the interaction and influence between text and language

*International Journal of*
*Computational Linguistics and Chinese Language Processing*
*vol. 19, no. 1, March 2014*

# Contents

**Papers**

Computational Linguistics and Chinese Language Processing
Vol. 19, No. 1, March 2014, pp. 1-10

# A Novel Approach for Handling Unknown Word Problem in Chinese-Vietnamese Machine Translation

Phuoc Tran*, and Dien Dinh+

## Abstract

For languages where space cannot be a boundary of a word, such as Chinese and Vietnamese, word segmentation is always the task to be done first in a statistical machine translation system (SMT). The word segmentation increases the translation quality, but it causes many unknown words (UKW) in the target translation. In this paper, we will present a novel approach to translate UKW. Based on the meaning relationship between Chinese and Vietnamese, we built a model which based on the meaning of the characters forming the UKW before translating the UKW through the model. Experiments show that our method significantly improved the performance of SMT.

**Keywords:** Chinese-Vietnamese SMT, Unknown Word, Sino-Vietnamese, Pure-Vietnamese, SVBUT Model, PVBUT Model.

## 1. Introduction

Unlike Western languages (typically English), Chinese and Vietnamese words are not separated by a space. A Chinese sentence consists of a series of characters, including punctuation, and no spaces between the characters. In Vietnamese, the spelled words (one-syllabled word) are separated by only one space, and the punctuation is located after the spelled words. Therefore, word segmentation is always solved first in Chinese or Vietnamese statistical machine translation (SMT) into other languages. The word segmentation increases the translation quality but generates many unknown words (UKW).

A Chinese word usually includes many meaningful characters; when translating it into Vietnamese, its meaning is usually divided into three cases. The first case is where the meanings of Chinese characters are their Sino-Vietnamese meanings, usually a 1-1 correspondence. The second case is where the meanings of the Chinese characters are similar

* Faculty of Information Technology, University of Food Industry, Ho Chi Minh City, Vietnam
E-mail: phuoctt@cntp.edu.vn

+ Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
E-mail: ddien@fit.hcmus.edu.vn

or related to the meaning of the Chinese word containing those characters. The final case is where the meanings of Chinese characters are not relevant to the meaning of the Chinese word containing them.

In the first case, Vietnamese words largely are borrowed from Chinese words (often called Sino-Vietnamese, which make up about 65% of the total number of Vietnamese words). Thus, the Sino-Vietnamese words generally appear in Vietnamese text. This very important feature is the basis for our handling UKW approach. In the second case, the meaning of the Chinese word is a combination of Pure-Vietnamese meanings of Chinese characters that form the Chinese word. For these two cases, we re-split a Chinese UKW into characters and translate the characters into Sino-Vietnamese or Pure-Vietnamese. Then, we proceed to incorporate the meanings of the characters and filter their meanings to be suitable to Vietnamese meaning.

In the final case, the meaning of Chinese word is not related to the meanings of the characters forming them. Named entity is a fairly common type of this case. In Chinese-Vietnamese SMT, a Chinese named entity is usually translated into its Sino-Vietnamese. Therefore, for these UKW, we will translate them into Sino-Vietnamese. Maybe the translation result is still not correct, but the quality is better than the previous translation, because the UKW are likely named entities.

This paper is presented as follows: in Section 2, we present related work. Our approach for handling UKW will be presented in Section 3. Meanwhile, in Section 4, we will present experiments and some discussion. Our conclusion will be presented in Section 5.

## 2. Related Work

Currently, there are many studies with different approaches to handle UKW to improve machine translation performance. Based on word's cognates and logical analogy, Joao *et al.* (2012) proposed two methods (cognates' detection and logical analogy) to translate UKW.

Another handling UKW approach was conducted by Matthias *et al.* (2008). The authors looked for the definition of the UKW in the source language and translated the definition (instead of translating the UKW). The definitions of UKW were automatically extracted from online dictionaries and encyclopedias and they were translated through the SMT system. The translation result would replace the UKW in the previous translation.

On the other hand, Zhang *et al.* (2008) translated Chinese UKW by re-splitting UKW into sub-words and translating the sub-words (sub-word based translation). Sub-word is a unit in the middle of a character and word. In addition, the authors also found that the quality of translation would increase significantly if applying NER to translate the UKW before using the sub-word based translation. Our approach is similar to this approach. Nevertheless, instead of re-splitting UKW into sub-words (greater than character), we re-split UKW into single characters and find their Sino-Vietnamese or Pure-Vietnamese meanings.

## 3. Chinese Character Meaning based UKW Translation Model

A Chinese UKW is re-translated by our model as follows.

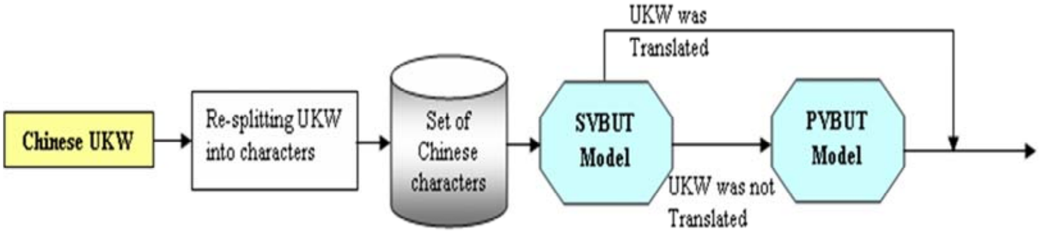


***Figure 1. Chinese character meaning-based UKW translation model.***

First, a Chinese UKW is disintegrated into Chinese characters before these characters are handled by the SVBUT model. Through this model, the UKW may be translated or not. If the UKW still has not been translated, it will continue to be translated by the PVBUT model. The two models will be presented in Section 3.1 and Section 3.2, respectively.

### 3.1 SVBUT Model (Sino-Vietnamese based Unknown Word Translation Model)

#### 3.1.1 About Sino-Vietnamese

Chinese, even in China, is pronounced differently, depending on the area, because there are many different voices or pronunciations, such as Cantonese, Hokkien and Beijing (Mandarin). Neighboring countries also have their own reading of Chinese, such as Korea having Sino-Korean (汉朝), Japanese having Sino-Japanese (汉和), and the Vietnamese having Sino-Vietnamese (汉越). Thus, Sino-Vietnamese is the reading way of Vietnamese people. For example, the Chinese word 银行 (bank) is pronounced “yín háng” (rendered using Pinyin), with the Vietnamese’s pronunciation being “ngân hàng”. A Chinese character may be pronounced by many Sino-Vietnamese words, but in a specific context, one Chinese character only corresponds to one Sino-Vietnamese. As in the above example, 银行, the corresponding Sino-Vietnamese pronunciation of character 银 is “ngân” and the pronunciation of 行 is “hành” “hạnh” “hàng” “hạng”. Nevertheless, when 银 and 行 are combined into the unique word, 银行, we only pronounce it “ngân hàng”.

### 3.1.2 SVBUT Model

Based on the meaning relationship between Chinese and Sino-Vietnamese, we built a novel model to translate UKW as follows.

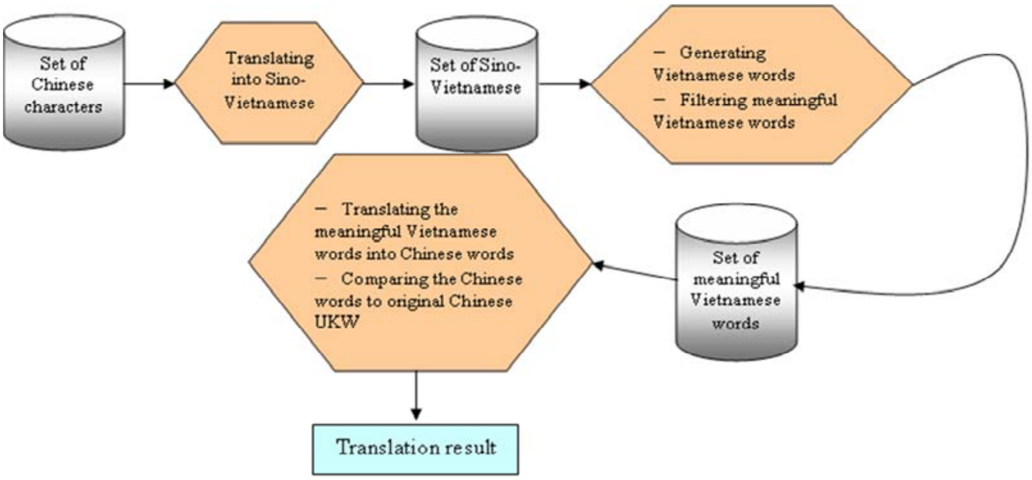
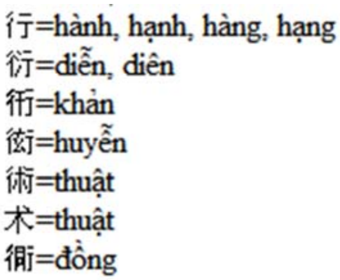

Set of Chinese characters
Translating into Sino-Vietnamese
Set of Sino-Vietnamese
– Generating Vietnamese words
– Filtering meaningful Vietnamese words
Set of meaningful Vietnamese words
– Translating the meaningful Vietnamese words into Chinese words
– Comparing the Chinese words to original Chinese UKW
Translation result

***Figure 2. SVBUT model***

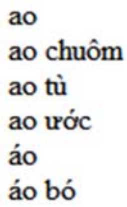
- Step 1: Translating the Chinese characters into Sino-Vietnamese. Based on a Sino-Vietnamese lexicon (Figure 3), we list all Sino-Vietnamese words of Chinese characters. A Chinese character may have many different Sino-Vietnamese words, but in a specific context, one Chinese character corresponds to one Sino-Vietnamese.



行=hành, hạnh, hàng, hạng
衍=diễn, diên
衎=khản
衒=huyễn
術=thuật
术=thuật
衕=đồng

***Figure 3. Sino-Vietnamese lexicon format***

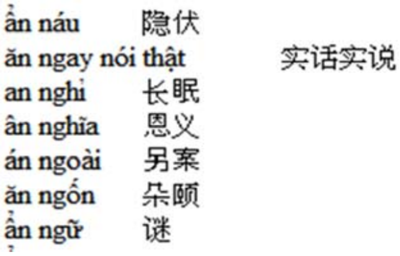
- Step 2: Generate a set of Vietnamese words from the Sino-Vietnamese words in Step 1. The generated Vietnamese words are formed by combining Sino-Vietnamese words together in the correct order in the source language. Then, based on a monolingual Vietnamese dictionary, we carry out filtering of the Vietnamese words, just using the meaningful Vietnamese words. The monolingual Vietnamese dictionary includes Pure-Vietnamese words and loanwords (mainly Sino-Vietnamese words). The format of the dictionary is presented in Figure 4.



ao
ao chuôm
ao tù
ao ước
áo
áo bó

***Figure 4. Monolingual Vietnamese dictionary format***

- Step 3: One Chinese word usually has one meaningful Sino-Vietnamese word and that is the meaning of the Chinese UKW. In case there are many meaningful generated Vietnamese words from one Chinese UKW, based on the Vietnamese-Chinese dictionary (Figure 5), we will look up the Chinese words corresponding to those Vietnamese words and compare them with the original Chinese UKW. If the Vietnamese word has a Chinese word that is the same as the Chinese UKW, it is the meaning of the UKW and it replaces the UKW in the translation results. If there are many meaningful Vietnamese words without any corresponding Chinese words, we will select the first word in a set of meaningful Vietnamese words to be meaning of Chinese UKW. Finally, if all generated Vietnamese words are meaningless, we will translate this Chinese UKW by the PVBUT model (Section 3.2).



| | |
|---|---|
| ẩn náu | 隐伏 |
| ăn ngay nói thật | 实话实说 |
| an nghỉ | 长眠 |
| ân nghĩa | 恩义 |
| án ngoài | 另案 |
| ăn ngốn | 朵颐 |
| ẩn ngữ | 谜 |

***Figure 5. Vietnamese-Chinese dictionary format***

For example, consider 银行 as a Chinese UKW; it will be translated through the SVBUT model as follows.

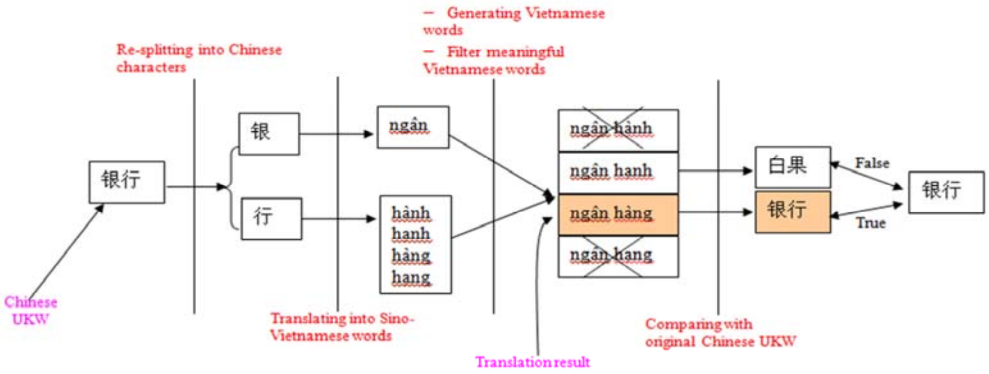

– Generating Vietnamese words
– Filter meaningful Vietnamese words
Re-splitting into Chinese characters
银行
银
行
ngân
hành
hanh
hàng
hang
ngân hành
ngân hanh
ngân hàng
ngân hang
白果
银行
False
True
银行
Chinese UKW
Translating into Sino-Vietnamese words
Translation result
Comparing with original Chinese UKW

***Figure 6. Chinese UKW 银行 is translated through SVBUT model.***

The Chinese UKW 银行 includes two characters, 银 and 行. 银 has a corresponding Sino-Vietnamese word “ngân” and 行 has four Sino-Vietnamese words, these are “hành” “hạnh” “hàng” “hạng”. Combining them together, we have four corresponding generated Vietnamese words. In these words, there are only two words that are meaningful, which are “ngân hàng” and “ngân hạnh”. Since “ngân hạnh” is a fruit type that is translated into Chinese to be “白果” we exclude the Vietnamese word because its Chinese word does not suit the original UKW. The remaining word “ngân hàng” (bank) has a corresponding Chinese word that is also Chinese UKW, so “ngân hàng” is chosen to be the meaning of the UKW 银行.

## 3.2 PVBUT Model (Pure-Vietnamese based Unknown Word Translation Model)

### 3.2.1 About Pure-Vietnamese

Vietnamese vocabulary, apart from words borrowed from other languages (mainly from Sino-Vietnamese words), is called Pure-Vietnamese. The word “Pure” in “Pure-Vietnamese” means vernacular (the native language). A Chinese character is often translated into a one-syllable Vietnamese word, and the few remaining can be translated into a Vietnamese word with more syllables. Some examples are 天/trời (heaven), 地/đất (land), 市/thành_phố (city). Another feature of the translation from Chinese to Pure-Vietnamese is that the meaning of the Chinese characters can be reorder in the Pure-Vietnamese translation. For example, the Chinese word 零钱 with 零/lẻ (loose) and 钱/tiền (cash, money), it is translated into Vietnamese as “tiền lẻ” (loose cash) (instead of “lẻ tiền”).

### 3.2.2 PVBUT Model

Based on the relationship of meaning between the Chinese and their Pure-Vietnamese, we built a UKW translation model as follows:

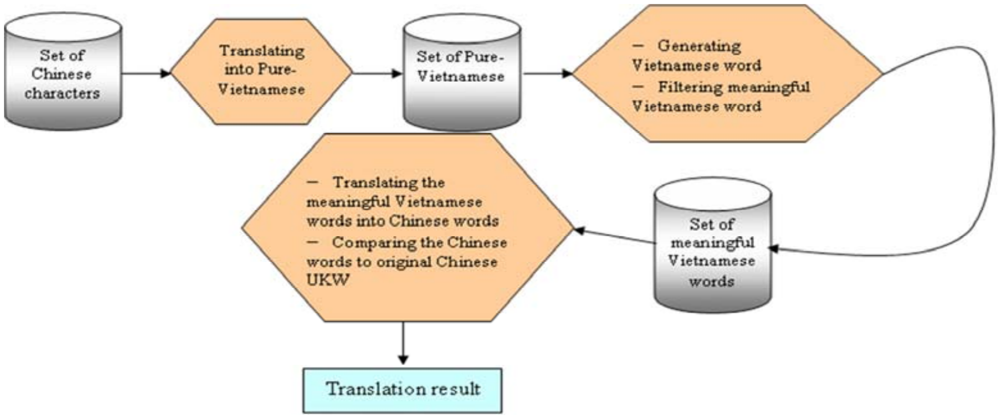

Set of Chinese characters
Translating into Pure-Vietnamese
Set of Pure-Vietnamese
– Generating Vietnamese word
– Filtering meaningful Vietnamese word
Set of meaningful Vietnamese words
– Translating the meaningful Vietnamese words into Chinese words
– Comparing the Chinese words to original Chinese UKW
Translation result

*Figure 7. PVBUT model.*

The PVBUT model is similar to SVBUT but there are some expansions. In Step 1, the meaning of a Chinese character can be a multi-syllabic word. In Step 2, the generated Vietnamese words, apart from the words being formed according to the order in the source language, must also include the words being established by reordering Vietnamese words that translated from the Chinese characters. The generated words will be filtered like in the SVBUT model.

After this period, the collection of meaningful Vietnamese words may not have any elements, may also have one element, or may have two elements or more. In the case where there is no element, we will translate the UKW as the Sino-Vietnamese (assuming the UKW to be a named entity). For the case of one element, the generated Vietnamese word is the meaning of the UKW. In the other case, where there is more than one meaningful element, we will select the first element in this collection to be the UKW's meaning. For example, Chinese UKW 零钱 will be translated by PVBUT model as follows.

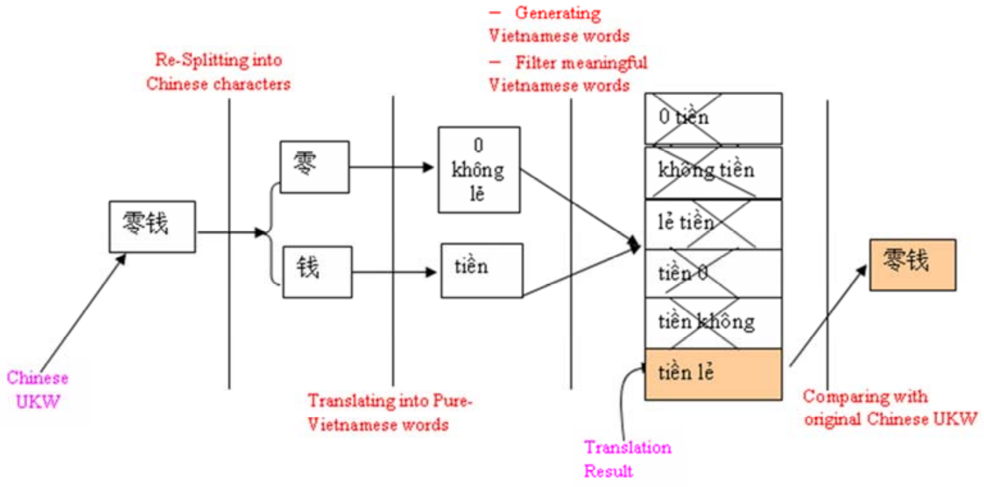

– Generating Vietnamese words
– Filter meaningful Vietnamese words
Re-Splitting into Chinese characters
零钱
零
钱
0
không
lẻ
tiền
0 tiền
không tiền
lẻ tiền
tiền 0
tiền không
tiền lẻ
零钱
Chinese UKW
Translating into Pure-Vietnamese words
Translation Result
Comparing with original Chinese UKW

***Figure 8. Chinese UKW 零钱 is translated through PVBUT model.***

The Chinese UKW 零钱 has two characters 零 and 钱. 零 has three Pure-Vietnamese meanings, which are "0" (zero), "không" (not) and "lẻ" (loose); 钱 has a common meaning of "tiền" (cash, money). Combining the Pure-Vietnamese meanings together, including reordering them, we get six generated Vietnamese words. In these six words, there is only "tiền lẻ" (loose cash) that is a meaningful Vietnamese word, the generated word "không tiền" (no money) is meaningful but it is not a Vietnamese word (it is a Vietnamese phrase). Fortunately, the word "tiền lẻ" has a corresponding Chinese word that is also an original UKW, so it replaces for the UKW in the final translation.

## 4. Experiments

Our experiment bilingual corpus consists of 20,000 Chinese-Vietnamese sentence pairs, which were extracted from Chinese conversational textbooks and online Chinese-Vietnamese forums, such as: "Textbook of 301 sentences in Chinese Conversation, Beijing Language Institute" and "Learning Chinese online, www.dantiengtrung.com.vn". Documents in the corpus are mostly communication text, so the length of the sentences is relatively short, with an average of about 10 words in a sentence. We use 90% of the sentences to train, 5% of sentences to test, and the remaining 5% of the sentences to develop. The training corpus (sentences to rain and developing) was trained by Moses[1] tool with the default parameters (SMT Baseline). We performed three experiments, Baseline translation, word segmentation translation, and translating UKW, by our model.

In the Baseline system, we considered the Chinese characters and the Vietnamese spelling words as the meaningful independent units. We inserted one space between Chinese characters and inserted one space between spelled words with the punctuation.

In the word segmentation system, we segmented Chinese words by the Stanford Chinese Segmenter tool[2]. This tool was installed by the CRF method (Conditional Random Field). For Vietnamese, we segmented words by our group's word segmentation tool. The segmenter was implemented by Dinh Dien *et al*. (2006), according to the Maximum Entropy approach.

Based on the results in the segmentation translation, we translated the sentences containing the UKW by our model. The BLEU score for each cases as follows.

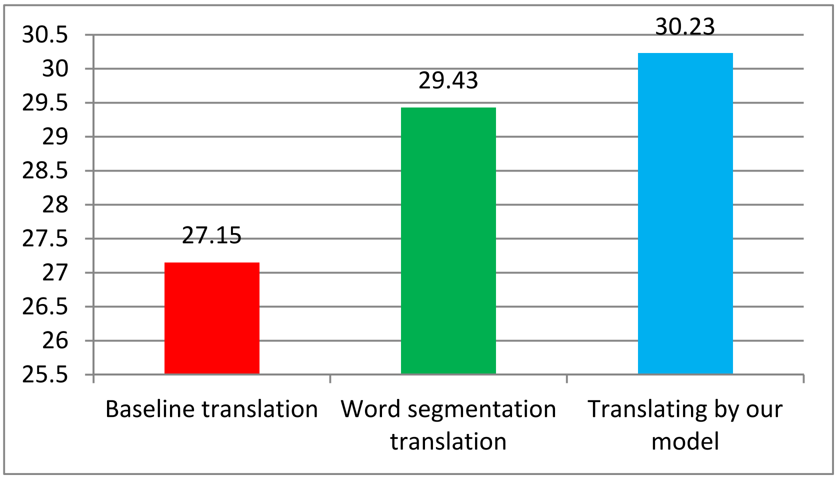


***Figure 9. Experiment results.***

[1] http://www.statmt.org/moses/

[2] Download: http://nlp.stanford.edu/software/segmenter.shtml

In the Baseline system, although it does not generate UKW, but it gives wrong result. For the case of word segmentation translation, its translation result is better than the Baseline's, but it generates many UKWs. The UKWs are translated through our system. The translation result shows that our system's translation quality is better than the Baseline system, as well as the word segmentation system. Here are two specific cases:

***Table 1. Two specific cases***

| ID | Chinese | True Translation | Baseline Translation | Word Segmentation Translation | Our Model |
|---|---|---|---|---|---|
| 1 | 假使 | Giả sử, nếu (if) | Kỳ nghỉ làm cho (holiday make) | 假使 | Giả sử (if) |
| 2 | 地点 | Địa điểm (location) | Địa giờ (land hour) | 地点 | Địa điểm (location) |

In both cases, the Baseline system did not generate UKWs but it gave wrong results. In the first case, the Chinese word 假使 includes character 假/ "kỳ nghỉ" (holiday) (in 放假 -> "nghỉ phép" (holiday)) and 使/ "làm cho" (make). Therefore, 假使 was translated "kỳ nghỉ làm cho" (holiday make). This result is completely wrong. A similar explanation can be seen for the second case.

For the word segmentation translation system, because the system did not recognize the Chinese words, it could not translate them and generated UKW. The UKW were translated by our model. In both cases, the meaning of UKW was also their Sino-Vietnamese meaning. Therefore, the UKWs were translated successfully by the SVBUT model.

In addition, to clarify the improvement of our model, we computed the Precision of the re-translation of UKW. Based on the word segmentation result, we selected 100 sentences containing UKW. Since the documents in the corpus are mostly communication texts, the length of each sentence is an average of about 10 words. Moreover, after segmenting words, the number of words in a sentence is less than 10 words. They were translated by MOSES; if there were UKW, each sentence often had only one UKW. Thus, in this paper, we only chose the sentences containing one UKW for precision calculation. We calculated the precision by the following formula:

$$\mathrm{Precision} = \frac{\sum \mathrm{Correct\ Pairs}}{\sum Total} \qquad (1), \text{Total} = 100 \text{ in this case.}$$

The 100 sentences were re-translated through our system. The system translated exactly 83 UKWs, gaining 83%. The remaining UKWs were translated into Sino-Vietnamese words. These words have no meaning in Vietnamese and also are not person names, place names, or organization names (these names are usually translated into Sino-Vietnamese). UKW 好的 is a specific case. The Sino-Vietnamese of this UKW is "hảo" "đích" and its Pure-Vietnamese is "tốt" (good), "của" (of). Both of "hảo đích" as well as "tốt của" are not Vietnamese words, so that our system will choose Sino-Vietnamese "hảo đích" to be the translation of UKW 好的. This result is completely wrong. We accept this incorrectness with perspective: a mistranslated result is not worse than a UKW result.

## 5. Conclusion

In this paper, we propose a novel approach to handle UKW in Chinese-Vietnamese SMT. This approach bases on meaning relations between Chinese and Vietnamese, including the relations between Chinese and Sino-Vietnamese and between Chinese and Pure-Vietnamese. The experiments show that our approach has significantly improved Chinese-Vietnamese SMT performance.

### Acknowledgement

This paper was performed under the sponsorship of NAFOSTED Fund and Kim Tu Dien Multilingual Data Center.

## References

Dinh, D., & Vu, T. (2006). A maximum entropy approach for Vietnamese word segmentation. In *Research, Innovation and Vision for the Future, 2006 International Conference on*, Ho Chi Minh, Vietnam, 248-253.

Eck, M., Vogel, S., & Waibel, A. (2008). Communicating Unknown words in machine translation. In *International Conference on Language Resources and Evaluation, LREC 2008*, Marrakech, Marocco.

Silva, J., Coheur, L., Costa, A., & Trancoso, I. (2012). Dealing with unknown words in the Statistical machine translation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, 3977-3981.

Tran, P., & Dinh, D. (2012). Surveying word boundary factor in Chinese-Vietnamese SMT. In *8th Science conference (HCMC University of Science, 2012)*, Ho Chi Minh, Vietnam.

Tran, P., & Dinh, D. (2012). Identifying and reordering prepositions in Chinese-Vietnamese machine translation. *First International Workshop on Vietnamese language and speech processing (VLSP), In conjunction with 9th IEEE-RIVF conference on Computing and Communication Technologies (RIVF 2012)*, Ho Chi Minh, Vietnam.

Zhang, R., & Sumita, E. (2008). Chinese Unknown word Translation by Sub-word Re-segmentation. In *International Joint Conference on Natural Language Processing*, Hyderabad, India.

Computational Linguistics and Chinese Language Processing
Vol. 19, No. 1, March 2014, pp. 11-32

# Joint Learning of Entity Linking Constraints Using a Markov-Logic Network

**Hong-Jie Dai[*], Richard Tzong-Han Tsai[+], and Wen-Lian Hsu[#]**

## Abstract

Entity linking (EL) is the task of linking a textual named entity mention to a knowledge base entry. Traditional approaches have addressed the problem by dividing the task into separate stages: entity recognition/classification, entity filtering, and entity mapping, in which different constraints are used to improve the system's performance. Nevertheless, these constraints are executed separately and cannot be used interactively. In this paper, we propose an integrated solution to the task based on a Markov logic network (MLN). We show how the stage decision can be formulated and combined in an MLN. We conducted experiments on the biomedical EL task, gene mention linking (GML), and compared our model's performance with those of two other GML approaches. Our experimental results provide the first comprehensive GML evaluations from three different perspectives: article-wide precision/recall/F-measure (PRF), instance-based PRF, and question answering accuracy. This paper also provides formal definitions of all of the above EL tasks. Experimental results show that our method outperforms the baseline and state-of-the-art systems under all three evaluation schemes.

**Keywords:** Entity Linking, Entity Disambiguation, Markov Logic Network, Gene Normalization

## 1. Introduction

Developing a system that can identify entities, such as personal names and gene or disease mentions, and that can classify the relations between them is useful for several applications in natural language processing and knowledge acquisition. There are several possible uses for

[*] Graduate Institute of BioMedical Informatics, Taipei Medical University
E-mail: hjdai@tmu.edu.tw

[+] Dep. of Computer Science & Information Engineering, National Central University
E-mail: thtsai@csie.ncu.edu.tw

[#] Institute of Information Science, Academia Sinica
E-mail: hsu@iis.sinica.edu.tw

such a system in different fields, *e.g.*, improving document retrieval for specific entities, relation extraction, and attribute assignment (*e.g.*, gene ontology annotations). In these applications, recognized entities must be linked to unique database entries. McNamee and Dang (2009b) named the task of matching a textual entity mention to a knowledge base (KB) entry *Entity Linking* (EL). In Figure 1, we provide a biomedical abstract to illustrate this task. The abstract discusses the relationship of the gene “CD59” to other lymphocyte antigens.

**TITLE:** Structure of the **CD59**-encoding gene: further evidence of a relationship to *murine* lymphocyte antigen Ly-6 protein

**ABSTRACT:** The gene for **CD59** [**membrane inhibitor of reactive lysis** (**MIRL**), **protectin**], a phosphatidylinositol-linked surface glycoprotein that regulates the formation of the polymeric **C9 complex** of complement and that is deficient on the abnormal hematopoietic cells of *patients* with paroxysmal nocturnal hemoglobinuria, consists of four exons spanning 20 kilobases. … PMID [1381503]

***Figure 1. An example of entity linking.***

After EL, the gene mention “CD59” in the first sentence must be linked to ID 966 in the Entrez Gene database of PubMed. In the first sentence, the authors also listed other designations of the gene, including “membrane inhibitor of reactive lysis” and “protectin,” and they defined “MIRL” as the abbreviation for “membrane inhibitor of reactive lysis.” Linking these instances to the same entry is a problem related to the *name variations* issue. Furthermore, the gene “CD59” may exist in multiple species. For example, it appeared in the title of the abstract as a murine gene, but turns out to be referring to a human (patient) gene in the first sentence. Therefore, each gene must be linked to its own unique database entry. Since these instances are polysemous, they are considered *entity ambiguity* issues. Finally, the “C9 complex” in the first sentence is a protein complex, but the Entrez Gene database does not contain this type of entity. When an entity cannot be associated with any entries, it is called an *absence* issue (McNamee & Dang, 2009b), and those entities are referred to as “*Nils*”.

Of all of the aforementioned issues, entity ambiguity is the most crucial problem (Dredze *et al*., 2010). Take the name “TP53” as an example. In the Entrez Gene database, there are over 300 proteins within over 20 species possessing the same name. Several disambiguation approaches have been proposed to address the problem. For example, Dredze *et al*. (2010) formulated the disambiguation task as a ranking problem and developed features to link entities to Wikipedia entries. Zhang *et al.* (2010) used an automatically generated corpus to train a binary classifier to reduce ambiguities. Dai *et al*. (2010) collected external knowledge for each entity and calculated likelihoods stating the similarity of the current text with the knowledge to improve the disambiguation performance.

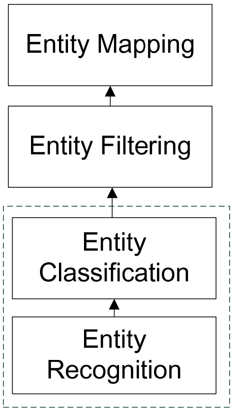


***Figure 2. Stages in the bottom-up EL approach: Some works combine the entity recognition and the entity classification into one step.***

Usually, a real-world EL system is constructed in a bottom-up manner, so it is necessary to make several decisions in different stages during the EL process. Figure 2 depicts the bottom-up process (Krauthammer *et al.*, 2004). *Entity recognition* marks single words (or several adjacent words) that indicate the presence of entities. As entity recognition does not determine the specific meaning of a concept, it is often combined with *Entity Classification*, which assigns entities to different classes, such as persons, genes, or diseases. After removing Nils (*Entity Filtering*), *Entity Mapping* maps entities to controlled database entries by calculating the similarities between the recognized entities and lexicon resources. This stage may resolve the entity ambiguity issue by a disambiguation process that uses contextual information to link entities to KB entries.

As shown in Figure 2, the traditional method for dealing with Nils has been to employ an additional step to filter out entities that have no corresponding entry in a KB. For example, Bunescu *et al*. (2006) filtered out mentions whose confidence scores are less than a fixed threshold. J Hakenberg *et al*. (2008) and Li *et al*. (2009) trained separate binary classifiers to validate linked mentions. Dredze *et al*. (2010) treated Nils as another KB entry candidate to train their EL ranking model.

Unfortunately, the separate-stage approach ignores possible dependencies among these stages and can result in error propagation. Continuing our example in Figure 1, in the EL stage, “MIRL” can be unambiguously linked to ID 996 with high confidence, because a search for the name in Entrez Gene returns only one match. Nevertheless, linking other mentions (*e.g.* “CD59” and “protectin”) to ID 996 is not as easy, since “CD59” alone has 18 candidate entries. These names can be linked with more ease when considered as synonyms of MIRL. Nevertheless, a divergent filtering stage may filter out the entity mention “MIRL” because it is listed as an abbreviation of organization names, such as Mineral Industry Research Laboratory.

With a joint inference process, we can carry out both tasks simultaneously to avoid this type of error propagation (Poon *et al.*, 2007).

Joint inference has become popular recently, because it allows features and constraints to be shared among different tasks. For example, J. R. Finkel *et al.* (2009) integrated parsing and named entity recognition into a joint model, whereas Dai *et al.* (2011) created a joint model for co-reference resolution and gene normalization and Liu *et al.* (2012) conducted entity recognition and normalization jointly for tweets. In this paper, we use the Markov Logic Network (MLN) (Richardson *et al.*, 2006), a joint model that combines first order logic and Markov networks, to capture the bottom-up decisions derived from the process illustrated in Figure 2. This model captures the contextual information of the recognized entities for entity disambiguation, as well as the constraints used when linking an entity mention to a database entry. For example, an entity mention can only be linked to a database entry when the mention has not been recognized as a Nil.

Existing EL evaluation metrics assess a system’s performance in terms of the effectiveness of database curation (Morgan *et al.*, 2008) or question answering (QA) accuracy (McNamee, Dang, *et al.*, 2009). In addition, we evaluate our system at a fine-grained entity by entity level. Such evaluation is more relevant to information extraction tasks, such as the bio-molecular event extraction task (Kim *et al.*, 2009).

When considering EL tasks from the entity level, one challenge is the lack of contextual information for disambiguating each individual entity. The major scheme of traditional entity disambiguation approaches relies on domain knowledge derived from entities’ profiles and contextual features extracted within a predefined content window. Rule-based (Dai *et al.*, 2010; Jörg Hakenberg *et al.*, 2008), vector space models (Cucerzan, 2007), and machine learning approaches (Crim *et al.*, 2005; Mihalcea *et al.*, 2007; Milne *et al.*, 2008) have been proposed to disambiguate entity mentions individually. Nevertheless, the context is unclear under certain circumstances. Take the sentence “The synthetic replicate of **urocortin** can bind with high affinity to Type 1 and Type 2 CRF receptors” as an example. The sentence itself does not explicitly provide any clues to help computer programs determine the identity of the gene mention “urocortin”, which has at least eight ambiguous Entrez Gene IDs. One approach is to expand the context window used for disambiguation to the paragraph level. Nevertheless, a paragraph described in a biomedical article usually incorporates several pieces of information in its description, which may not be related directly to a target entity instance and leads to the failure of traditional EL approaches.

Our idea of dealing with the challenge of deficient contextual information for disambiguating individual entity instances is to model dependencies among entities across sentences in the same paragraph. These dependencies have been ignored by most of the previous EL approaches. We refer to our approach as the collective EL, which is developed by considering the relational information hidden among entities. In the following sections, we first give formal definitions of the EL tasks mentioned above, followed by an introduction of MLN and a description of the main ideas of the proposed EL method with the formulation of the collective EL approach.

## 2. Entity Linking Problem Definition

This section gives formal definitions of all related EL tasks.

**Definition 1: Instance-based Entity Linking Problem**

Let $M = (m_1, m_2, \ldots)$ denote a sequence of entities mentioned in an article $A$. The surface name of $m_i$ is denoted by *Name*($m_i$). The named entity type of $m_i$ is *EntityType*($m_i$). The surrounding context of $m_i$ can be extracted by *Context*($m_i$). Given a KB containing a set of entries $ID = \{id_1, id_2, \ldots\}$, each of which organizes knowledge related to an entity, the instance-based EL problem is defined as finding a mapping function *LinkTo*($m_i$) that maps each $m_i$ in $M$ to a unique entry $id_i$ in $ID$ and satisfies the constraint $\left|LinkTo(m_i) : m_i \in M\right| = \left|M\right|$.

In instance-based gene mention linking (GML), only entities whose *EntityType*($m_i$) belong to "gene" are considered for evaluation. Both the gene normalization task in BioCreAtIvE (Morgan *et al*., 2008) and the EL task in the KB population (McNamee & Dang, 2009a) can be subsumed into Definition 1. In BioCreAtIvE gene normalization, the developed system should satisfy the equation $\left|LinkTo(m_i) : m_i \in M\right| \le \left|M\right|$. We refer to this task as the article-wide EL problem.

**Definition 2: Article-wide Entity Linking Problem**

Let $M = \{m_1, m_2, \ldots\}$ denote a set of entities mentioned in $A$. Given the entries $ID = \{id_1, id_2, \ldots\}$ in a KB and the mapping function *LinkTo*($m_i$), the article-wide EL problem satisfies the constraint $\left|LinkTo(m_i) : m_i \in M\right| \le \left|M\right|$.

On the other hand, the KB population EL task only considers one certain entity $m_i$ mentioned in $A$. We refer to this task as the article-wide "salient entity" linking problem, in accordance with the Wikipedia style manual, in which only the salient entity and its related entities should be linked in wikification. Excessive links would obstruct the readers in following the article by drawing attention away from important links (Mihalcea & Csomai, 2007).

**Definition 3: Article-wide Salient Entity Linking Problem**

Let $M = m_i$ denote the salient entity mentioned in $A$. Note that, in encyclopedia-style articles, $\left|M\right| = 1$ because the same surface name described in such articles should refer to the same instance. Given the entry set $ID = \{id_1, id_2, \ldots\}$ of a KB, the purpose of the article-wide salient EL problem is to find the mapping function *LinkTo*($m_i$) that links $m_i$ to a unique entry $id_i$ in $E$.

Note that, in the KB population EL subtask (pertained to the article-wide salient EL problem), the salient entity is given. Nevertheless, in the instance-based GML or the BioCreAtIvE gene normalization (pertaining to the article-wide EL problem) tasks, the systems must also deal with the entity recognition/classification problem.

## 3. First-order Logic and Markov Logic Networks

Markov logic is a statistical relational learning language based on first-order logic (FOL) and Markov networks. In this section, we consider FOL and Markov networks in terms of the GML task.

In FOL, the formulae are constructed using four types of symbols: constants, variables, functions, and predicates. For GML, a constant symbol may represent a gene mention (*e.g.* "CD59") or its unique database entry (*e.g.* the Entrez Gene ID "966"). If variables and constants are type-specific, their range can only cover objects of the corresponding type. To give an example, the variable *y*'s range covers all Entrez Gene database IDs. Predicate symbols are used to represent the relations between *terms*; for example, we can define the predicate, *LinkTo*(*x*, *y*), to indicate that a gene mention (the variable *x*) should be linked to an entry (the variable *y*). Formulae are constructed recursively from predicates applied to a tuple of terms by through use of logical connectives and quantifiers. Then, we can model the EL task by introducing a set of logical predicates. For instance, we can define the predicate *Candidate*(*i*, *j*) to indicate that the gene mention *i* can be mapped to an entry *j*. The predicate captures information about gene mentions and their corresponding candidate database entries. Through this predicate, we can infer whether a gene mention is unambiguous. Then, we can use the following formula

**Formula 1:** $\forall x \exists ! id.Candidate(x, id) \Rightarrow LinkTo(x, id)$

to model the concept that, when a gene mention is mapped to only one entry, it should be linked to that entry. Note that we use the symbol, $\exists !$, to refer to a uniqueness quantification.

A first-order KB is a set of hard constraints on the set of ground atoms of predicates (or so-called *possible worlds*). If a world violates any formula, it has zero probability. In most domains, however, it is very difficult to derive non-trivial formulae that are always true. Markov logic softens these constraints to handle uncertainty by associating each formula with a weight that reflects the strength of a constraint. Ideally, if we could define a formula with a proper weight for its distribution, a world in which the formula is satisfied would have a higher probability than a world in which it is not. In Markov logic, a set of weighted formulae is called a MLN.

**Definition 4: Markov Logic Network**

An MLN *L* is a set of pairs ($F_i$, $w_i$), where $F_i$ is a formula in FOL and $w_i$ is a learned weight corresponding to the $F_i$ whose value is a real number. In combination with a finite set of constants $C = \{c_1, c_2, \ldots, c_{|C|}\}$, it defines a Markov network $M_{L,C}$ as follows: $M_{L,C}$ contains one node for each possible grounding of each predicate appearing in $L$. The value of the node is 1 if the ground predicate is true, otherwise it is 0.

Based on the definition, we can generate a graph structure of the ground Markov network where there is an edge between two nodes of $M_{L,C}$ if the corresponding ground atoms appear together in at least one grounding of one formula in $L$. Thus, the predicates in each ground formula form a clique in $M_{L,C}$. Each clique in the graph is associated with a potential function $\phi_i$ .The joint distribution of a set of variables $X$ represented by $M_{L,C}$ then is defined by:

$$\mathrm{P}(X = x) = \frac{1}{Z} \prod_i \phi_i (x_{\{i\}})^{n_i(x)}$$

where $x_{\{i\}}$ is the state of the $i$th clique (*i.e.*, the exact values of the predicates that appear in that clique $F_i$), $n_i(x)$ is the number of true groundings of $F_i$ in $x$, and $\phi_i(x_{\{i\}}) = e^{w_i}$ . $Z$ is the partition function given by $Z = \sum_{x \in X} \prod_i \phi_i(x_{\{i\}})$ . Markov networks are often represented as log-linear models in which each clique is replaced by an exponential weighted sum of the features of the state, leading to

$$\mathrm{P}(X = x) = \frac{1}{Z} \exp\left( \sum_i w_i f_i(x) \right)$$

In our implementation, $f_i$ is a binary feature, $f_i(x) \in \{0,1\}$ .

## 4. The Proposed Entity Linking System for Gene Mention Linking

Figure 3 illustrates the input of the proposed EL system developed for GML and the FOL predicates defined for the corresponding bottom-up stages. The input is an article, such as a biomedical abstract. The given article is first processed by a gene mention recognizer to identify gene mention boundaries, and the employed gene mention mapper then maps each recognized gene to a list of candidate identifiers, based on a lexicon compiled from the Entrez Gene database.

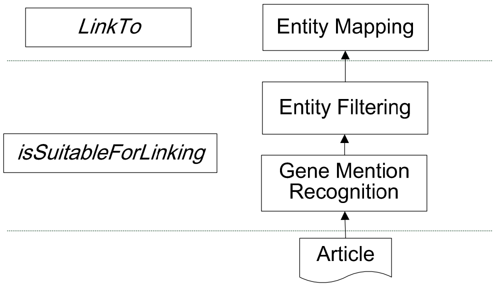


***Figure 3. MLN hidden predicates defined for each stage.***

Ideally, we should be able to treat all recognized gene mentions as candidates, and proceed directly to the entity mapping task. Nevertheless, the employed recognizer may generate false positive gene mentions. Such mentions can be classified into two types: those that do not belong to any entity class and those that belong to classes that are not the curation target (Nils). In GML, Nils appear when the gene mentions are DNA polymerases, or in a specific organism that is not considered. To capture the concept, we define the predicate *isSuitableForLinking*(*x*), which indicates that the gene mention *x* of the article is suitable for linking to an entry. For entity mapping, we use the predicate *LinkTo*(*x*, *id*) to represent that the gene mention *x* must be linked to the database entry *id*. As the objective of the entity mapping task is to determine a unique KB entry for each entity, we must define a formula to ensure that the constraint is satisfied. Regarding GML, we use the following formula to prevent a gene mention from associating with more than two identifiers.

**Formula 2: Entity Mapping Constraint**

$$LinkTo(x, id_i) \wedge id_i \neq id_j \Rightarrow \neg LinkTo(x, id_j)$$

## 4.1 Formulation of the Instance-based GML

Within the machine learning community, classification is typically done on each object independently without taking into account any underlying relation that connects the objects. In most of the individual EL formulations, an individual classifier is employed to assign a probability to the linked ID of an individual instance independently of the linked IDs of other instances. For example, the following formula expresses that, if the chromosome location information of the entity mention *x*, which has the KB entry *id* as its candidate ID, can be found in the surrounding text, *x* should be linked to *id*.

**Formula 3:** $hasChromosomeInfo(id) \wedge Candidate(x, id) \Rightarrow LinkTo(x, id)$

Other useful biographical information for locally disambiguating gene mentions includes tissues, gene ontology, and PPI. Some researchers (Hakenberg *et al*., 2007; Lai *et al*., 2009) have used this information for individual GML. Table 1 shows the observed predicates and formulae defined for this individual approach. We take *hasPPIPartnerRank* as another example. For the individual GML process, we define the following formula.

**Formula 4: Individual PPI**

$$\exists! id.hasPPIPartnerRank(x, id, 1) \wedge \exists w.hasWord(w) \wedge isPPIKeyword(w) \Rightarrow LinkTo(x, id)$$

implies that the gene mention *x* should be linked to the *id* that has the most PPI partners. The other predicates, including *hasGOTermRank* and *hasTissueRank*, follow a similar concept, in which the context is matched with the corresponding keywords to determine the frequency in the given abstract text.

***Table 1. Observed predicates and formulae defined for entity mapping.***

| | |
|---|---|
| | *Candidate*(*x*, *id*)<br>*hasChromosomeInfo*(*id*)<br>*hasWord*(*w*): the abstract contain a word *w*.<br>*isPPIKeyword*(*w*), *isPPIPartner*($id_1$, $id_2$)<br>*hasPPIPartnerRank* (*x*, *id*, *r*)<br>*hasGOTermRank*(*x*, *id*, *r)*<br>*hasTissueTermRank*(*x*, *id ,r)*<br>*hasDictionaryMatchRank*(*x*, *id*, *r*)<br>*hasPrecedingWord*(*x*, *w*, *l*), *hasFollowingWord*(*x*, *w*, *l*)<br>*hasUnigramBetween*(*x*, *y*, *w*) |
| Variable Type | *x*: integer that refers to the *x*th gene mention in the given article (similarly *y* refers to the *y*th gene mention)<br>*id*: an Entrez Gene ID, which refers to the linked KB entry.<br>*w*: a word.<br>*r*: integer that refers to the rank of the matching.<br>*l*: integer that refers to a context window length. |
| Formulae | $Candidate(x,id) \wedge hasChromosomeInfo(id) \Rightarrow LinkTo(x,id)$<br>$hasWord(w) \wedge PPIKeyword(w) \wedge Candidate(x,id) \wedge \exists! id_i.MostPPIPartners(id_i) \wedge id_i = id \Rightarrow LinkTo(x,id)$<br>$Candidate(x,id) \wedge \exists! id_i.MostGOTerms(id_i) \wedge id_i = id \Rightarrow LinkTo(x,id)$<br>$Candidate(x,id) \wedge \exists! id_i.MostTissueTerms(id_i) \wedge id_i = id \Rightarrow LinkTo(x,id)$<br>$\exists! id.Candidate(x,id) \Rightarrow LinkTo(x,id)$<br>$\exists! u,id.Candidate(x,id) \wedge has\Pr ecedingWord(x+1,u) \wedge u = "(" \wedge hasUnigramBetween(x,x+1,u) \wedge hasFollowingWord(x+1,")") \Rightarrow LinkTo(x+1,id)$<br>$\exists! u,id.Candidate(x+1,id) \wedge has\Pr ecedingWord(x+1,u) \wedge u = "(" \wedge hasUnigramBetween(x,x+1,u) \wedge hasFollowingWord(x+1,")") \Rightarrow LinkTo(x,id)$ |

A drawback of individual EL classifiers is that, when they decide the linking entry of an entity, they cannot utilize information about the linked entries and features of other entities in the same article. Nevertheless, those entity instances can be related, and the interrelationship can be used to improve the EL performance. Furthermore, there are strong dependencies among the unknown IDs of the instances, which could either be a true positive entity mention or a Nil. These dependencies are highly nonlocal.

*Collective classification* refers to the task of inferring labels for a set of objects using not just their attributes, but also the relations among them (Sen *et al.*, 2008).

**Definition 5: Collective Classification**

Given a network *N*, a node *n* in *N*, and the label set *L*, there are three distinct feature types that can be utilized to determine the label *l* of *n*, where $l \in L$.

1. The observed features of *n*.
2. The observed features (including observed labels if they are known) of nodes in the neighborhood (related nodes) of *n*.
3. The unobserved labels of nodes in the neighborhood (related nodes) of *n*.

In our formulation for EL, for a given article, the candidate database entries of all recognized entities form the network *N*. A mention's candidate entry and order form the node *n* = (*id*, *order*) in *N*, and an edge exists between two nodes if they have dependencies. In this work, the dependencies are constructed based on two main ideas: the discourse salience property in centering theory (Grosz *et al*., 1995) and the protein-protein interaction (PPI) association.

### 4.1.1 Discourse Salience

*Discourse salience* is a phenomenon where, in a given discourse, there is precisely one entity that is the center of attention. This entity is mentioned over and over again, which makes it more salient than others. We utilize this phenomenon to improve the instance-based EL confidence. Suppose that *id* is a candidate database entry for several entities in a discourse, we then can assume that *id* is more salient than other database entries. If the EL system can link one of these mentions to *id* with high confidence, then the system is more likely to be able to link all of the other mentions to *id* as well.

### 4.1.2 Protein-protein Interaction

Similarly, the idea of employing the PPI association allows us to express the concept that a gene mention *y* should be linked to $id_j$ if another gene mention *x* has been linked to $id_i$ and $id_i$ forms an interaction with $id_j$.

In order to capture the concepts above, the order of all individual instances described in an abstract are leveraged to build dependencies in our formulation. The lack of local contextual information then can be resolved by the constructed dependencies. In our work, the salience collective is written as follows in Markov logic.

**Formula 5: Salience collective**

$$Pr\,ecede(x, y) \wedge LinkTo(x, id) \wedge Candidate(y, id) \Rightarrow LinkTo(y, id)$$

If the database entry *id* is linked to an entity *x* that precedes the current mention *y* and *id* is a candidate entry of *y*, then the current entity *y* should also be linked to *id*. This formula is similar to the transition feature of the linear-chain conditional random fields (Lafferty *et al*., 2001), which can be implemented in Markov logic as follows.

**Transition feature:** $Pr\ e\ cede(x,y) \wedge Label(x,+L) \Rightarrow Label(y,+L)$

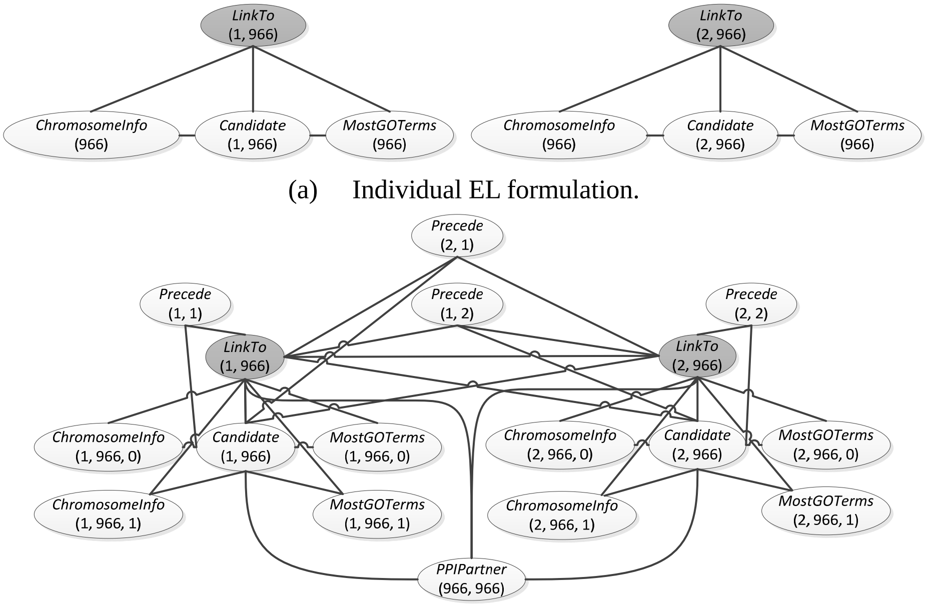
Note that the symbol "+" in the above formula directs the MLN learning algorithm to associate the formula with a different weight depending on variables containing the "+" notation.

We define the predicate *PPIPartner*($id_i$, $id_j$), whose value is true if $id_i$ and $id_j$ form a PPI pair. We then use the following formula to capture the PPI association concept.

**Formula 6: PPI**

$$LinkTo(x,id_i) \wedge Candidate(y,id_j) \wedge PPIPartner(id_i,id_j) \Rightarrow LinkTo(y,id_j)$$

Based on these two collective formulae, Figure 4 compares the ground Markov network (b) of our collective formulation with the traditional individual approach (a). In Figure 4 (a), the individual approach considers the likelihoods stating the similarity of the current context with the domain knowledge of the recognized entity, including chromosome location (*ChromosomeInfo*) and gene ontology (*MostGOTerm*). Comparing Figure 4 (b) with (a), our collective formulation captures the dependencies among entities, allowing the information to be employed in the GML decision.



(a) Individual EL formulation.

(b) Collective EL formulation.

***Figure 4. Ground Markov network obtained by applying example formulae to the constants x, y = {1, 2}, c = {0, 1}, and id = {966}.***

## 4.2 Formulae for Entity Recognition/Classification and Filtering

The hidden predicate *isSuitableForLinking* captures the decisions made after the entity recognition/classification stage. When the gene mention *x* is linked to an identifier *id*, we employ the following formula to ensure that it is an entity suitable for linking.

**Formula 7:** $LinkTo(x, id) \Rightarrow isSuitableForLinking(x)$

Note that the formula models the bottom-up decision, as shown in Figure 2. The identifier *id* does not have to be linked to the gene mention *x* proposed by the entity recognition/classification stage. Nevertheless, the *id* cannot be assigned to the gene mention *x* that has not been proposed as a potential entity.

Our first formula for *isSuitableForLinking* treats all gene mentions as potential entities:

$$hasName(x, n) \Rightarrow isSuitableForLinking(x) .$$

The other formulae are constructed using the observed predicates defined in Tables 1 and 2 to check the contextual information. For example,

$$hasFirstWord(x, +w) \wedge isSpeciesTerm(+w) \Rightarrow isSuitableForLinking(x)$$

implies that the suitability of a certain gene mention for linking depends on whether or not the first word is a keyword for a certain species.

***Table 2. Observed predicates for entity filtering.***

<table>
<tr><td colspan="2"><i>hasName</i>(<i>x</i>, <i>n</i>)<br><i>hasFirstWord</i>(<i>x</i>, <i>w</i>), <i>hasLastWord</i>(<i>x</i>, <i>w</i>)<br><i>hasPrefix</i>(<i>x</i>, <i>ch</i>, <i>d</i>, <i>l</i>): the <i>x</i>th gene mention has a prefix <i>ch</i> of length <i>l</i>, and the prefix’s case is the same as its following character (<i>d</i> = 0) or different (<i>d</i> = 1).<br><i>isSpeciesTerm</i>(<i>w</i>), <i>isAllUpperCase</i>(<i>i</i>), <i>hasPartOfSpeech</i>(<i>x</i>, <i>k</i>, <i>p</i>)<br><i>isContainedMoreSpecificMentions</i>(<i>x</i>)</td></tr>
<tr><td>Variable Type</td><td><i>n</i>: a word or a sequence of words that refer to the surface name of a gene mention.<br><i>ch</i>: a character.<br><i>d</i>: an integer.<br><i>k</i>: the <i>k</i>th index of the gene mention.<br><i>p</i>: a part-of-speech</td></tr>
</table>

# 5. Experimental Results and Discussion

## 5.1 Experimental Setup

### 5.1.1 Evaluation Metrics

We use three metrics to evaluate our approach and compare it with other GML methods. The first and second metrics used the standard precision, recall, and F-measure metrics (PRF) at two resolutions (article and instance).

Article-wide evaluation used the standard used in the BioCreAtIvE challenge (Hirschman *et al*., 2005), which is designed to determine an GML system's performance as an aid for the curation of biological databases. The GML system outputs a list of IDs for a given article, and this list is compared to the gold standard ID list. The PRF scores are calculated based on the sums of true/false positives/negatives (TP, TN, FP, FN).

Instance-based evaluation measures the GML performance at a fine-grained resolution (Dai *et al*., 2011). In contrast to the first metric, the PRF scores are calculated based on the sums of TP, TN, FP, and FN for all instances in the test dataset. Therefore, under this criterion, an FP can link a true gene mention to the wrong KB entry or link a false gene mention to any entry, while an FN can link a true gene mention to the wrong entry or fail to recognize a true gene mention. For TP/FP/FN, we need to determine when the predicted boundary matches that of the gold standard. Most pure entity recognition tasks use "exact-matching" as the primary criterion. Under this criterion, a candidate gene mention can only be counted as a TP if both its left and right boundaries fully coincide with the gold answer. In a real case, however, a gene mention can be tagged in several ways (*e.g.*, "serum $_{<entity>}$LH$_{</entity>}$" and "$_{<entity>}$serum LH$_{</entity>}$" are both correct), which are intrinsic to the annotation of any gene mention corpus, whether developed by humans or machines (Tsai *et al*., 2006), and may depend on the annotator's perspective (Franzén *et al*., 2002). Furthermore, for the GML task, the correctness of the linked entry is more important than its boundary. Therefore, we used the approximate-match to determine the boundary criterion. For example, a TP is counted when a machine-linked gene mention is a substring of the gold standard-linked gene mention or vice versa and the linked entry is equal to the gold entry.

The third metric, the mean accuracy across all queries, considers the QA perspective. EL is important in QA systems because the systems rely on data from multiple sources, so name ambiguity will lead to wrong answers and poor results. We adopt the evaluation metrics used in text analysis conference KB population (KBP) track 2009 (McNamee & Dang, 2009a) to report Accuracy$_{micro}$ and analyze the results from the QA perspective [1].

### 5.1.2 Datasets

In the experiments, we used the training and test sets (281 and 262 abstracts, respectively) released by the BioCreAtIvE gene normalization task (Morgan *et al*., 2008) for article-wide evaluation. The corpus contains annotations for human genes that are linked to IDs in the Entrez Gene database. Although the gold answers contain each ID's surface name, they do not give the exact location of the corresponding gene mention in the abstract. To obtain

[1] For details please refer to http://apl.jhu.edu/~paulmac/kbp.html.

fine-grained evaluation results, our in-lab biologists compiled an instance-based GML corpus by annotating the exact location and the boundary of the IDs' gene mentions (Dai *et al.*, 2012). After compiling the corpus, we performed three-fold cross validation (CV) on the training dataset to optimize the weights and formulae and to evaluate its performance on the test set.

In QA evaluation, as defined in KBP, the associated document is used to provide contextual information that might be useful for linking. We paired the surface names with their corresponding documents as the input query. For each query, the corresponding gold answer could be 1) an Entrez Gene ID or 2) a Nil in cases where our biologists annotated the entity as a mention without associating it with any IDs. Table 3 shows the generated query/answer pairs based on the BioCreAtIvE Corpus.

***Table 3. The statistics of the generated query/answer pairs on the BioCreAtIvE corpus.***

| Dataset | # of queries | # of Nil | # of Entities |
|---|---|---|---|
| Training | 1073 | 87 | 1132 |
| Test | 1266 | 66 | 1154 |

### 5.1.3 Model Configurations

To assess the performance of our models and determine the possible gains that can be achieved by considering a collective model and a joint model of the bottom-up stages, we designed two configurations. The first configuration was the collective model, which used all of the disambiguation formulae defined in Section 4.1 (denoted as **CM**). The constructed Markov network resembles Figure 4 (b) with additional grounding for the predicates and individual formulae defined in Table 1. The second configuration further included the formulae defined in Section 4.2 on CM to build a joint model (denoted as **JCM**). This work used the 1-best Margin Infused Relax online learning Algorithm (MIRA) (McDonald *et al.*, 2005) for learning weights and employed cutting plane inference (Riedel, 2008) with integer linear programming as its base solver for inference at test time as well as during the MIRA online learning process.

In addition, we compared the first configuration (CM) with two GML approaches: Lai *et al.* (2009)'s rule-based approach and Crim *et al.* (2005)'s maximum entropy (ME) approach, which handled the GML task as an individual classification problem. The ME approach was adopted with the individual features described in Section 4.1.

Finally, to compare the stage-based approach with the second configuration (JCM), which performs joint filtering and linking, we trained a separate ME model via the features designed for the *isSuitableForLinking* stage in Figure 3. This model then was combined with the first configuration, the Rule-based approach and ME approach, which we denote as $\text{CM}_{\text{stage}}$, Rule-based$_{stage}$, and ME$_{stage}$, respectively.

For the above configurations, we employed Lai *et al.*'s system[2] to recognize all gene mentions and generate mapped candidate IDs for each mention. All configurations were based on the same candidate ID sets.

In the next sub-section, we first discuss the fine-grained resolution results. Then, we derive BioCreAtIvE's evaluation results by simply merging the linked identifiers in all indices and removing duplicated identifiers. Finally, the results are displayed from the QA perspective.

***Table 4. The three-fold CV results on the training set using instance-based criterion. Our models are highlighted in bold.***

| Config. | P (%) | R (%) | F (%) | Diff (F) |
|---|---|---|---|---|
| No Disambiguation | 81.0 | 48.0 | 60.3 | - |
| Salience Collective | 79.9 | 49.6 | 61.3 | +1.0 |
| PPI Collective | 79.3 | 51.2 | 62.2 | +1.9 |
| Rule-based | 71.7 | 54.0 | 61.6 | +1.3 |
| ME | 79.8 | 48.9 | 60.5 | +0.5 |
| **CM** | 73.5 | 55.9 | 63.5 | +3.2 |
| Rule-based$_{stage}$ | 71.7 | 54.0 | 61.6 | +1.3 |
| ME$_{stage}$ | 86.4 | 46.9 | 60.8 | +0.8 |
| **CM$_{stage}$** | 73.5 | 55.9 | 63.5 | +3.2 |
| **JCM** | 79.9 | 54.9 | 65.1 | +4.8 |

## 5.2 Experiment Results

Table 4 shows the fine-grained results derived on the training set. The employed system's linking performance without applying any disambiguation approaches is shown in the first row (No Disambiguation.) Our model can simulate the same PRF scores when only Formula 1 is applied. The last column shows the improvement of F-score over the baseline after implementing different GML disambiguation methods.

It can be observed that, by adding the salience collective (Formula 5) without any disambiguation formulae and domain knowledge, the recall rate is improved by 1.6%, which results in an improved F-score. This demonstrates that a scientific article often contains repetitive information, such as key genes, which can be captured by the formula. Furthermore,

[2] The employed system can be downloaded from https://sites.google.com/site/potinglai/downloads.

the PPI collective combining the domain knowledge achieves a higher PRF-score, even outperforming the Rule-based and ME.

***Table 5. Results derived on the test set.***

| Metrics | Fine-grained Resolution (%) | | | | Aids for Curation (%) | | | |
|---|---|---|---|---|---|---|---|---|
| Config. | P | R | F | Diff | P | R | F | Diff |
| No Disambiguation | 80.7 | 56.3 | 66.3 | 0 | 77.3 | 71.5 | 74.3 | 0 |
| **Salience Collective** | 79.5 | 59.0 | 67.7 | +1.4 | 77.2 | 71.3 | 74.1 | -0.2 |
| Rule-based | 72.9 | 63.9 | 68.1 | +1.8 | 82.6 | 83.4 | 83.0 | +8.7 |
| ME | 79.2 | 58.2 | 67.1 | +0.8 | 88.8 | 79.0 | 83.6 | +9.3 |
| **CM** | 73.8 | 64.3 | 68.7 | +2.4 | 86.1 | 83.0 | 84.5 | +10.2 |
| Rule-based$_{stage}$ | 73.7 | 64.2 | 68.7 | +2.4 | 84.1 | 83.7 | 83.9 | +9.6 |
| ME$_{stage}$ | 80.2 | 58.4 | 67.6 | +1.3 | 90.2 | 79.0 | 84.3 | +10 |
| **CM$_{stage}$** | 74.3 | 64.3 | 69.0 | +2.7 | 87.9 | 83.2 | 85.5 | +11.2 |
| **JCM** | 77.5 | 63.7 | 69.9 | +3.6 | 87.7 | 83.8 | 85.7 | +11.4 |

The results derived on the test set by fine-grained resolution and aided for curation and QA are shown in Tables 5 and 6, respectively. In summary, we observe that the collective GML method consistently outperforms the compared methods under the three criteria. Moreover, the comparison of our joint model (JCM) and the stage models shows that the joint model performs better under all evaluation metrics.

***Table 6. QA results on the test set.***

| Config. | Accuracy |
|---|---|
| No Disambiguation | 65.7 |
| **Salience Collective** | 67.2 |
| Rule-based | 72.4 |
| ME | 66.8 |
| **CM** | 73.1 |
| Rule-based$_{stage}$ | 73.0 |
| ME$_{stage}$ | 67.0 |
| **CM$_{stage}$** | 73.3 |
| **JCM** | 73.5 |

We also observe that adding the salience collective reduces the recall rate in the aid for curation evaluation. According to our analysis, the salience collective improves the recall in the fine-grained evaluation. In contrast, for database curation, the collective tends to improve the overall precision. Nevertheless, it reduces the recall. By removing the salience collective from CM, the P improved by 0.8% but the R reduced by 0.7% in the test set. This phenomenon is reasonable because adding the dependency causes the model to link mentions with previous linked IDs.

Finally, the results also reveal the performance gap (approximately 15.8%) when we want to employ the GML system, which is evaluated in terms of the database curation criterion with 80+% F-score on advanced IE tasks, such as relation or event extraction.

## 5.3 Discussion

Evaluating EL from the fine-grained perspective allows us to analyze the task in detail. In this section, we describe our findings and propose potential research directions.

One advantage of employing MLN in our EL modeling is that it is easy to model arbitrary longer range dependencies, as expressed by Formula 5 and Formula 6. It is difficult to model such dependencies using ME. As shown in Tables 4 and 5, adding the collectives improves the fine-grained EL performance.

Another advantage of our GML approach is that it is flexible and can be applied quickly in real-world applications. The EL task usually is defined as linking a mention to a unique entry. Nonetheless, in the biomedical field, there are some mention descriptions that cannot be linked to unique IDs. The following are some examples extracted from our corpus:

1. **ABCB9 protein** appears to be most highly expressed in the Sertoli cells of the seminiferous tubules in *mouse and rat testes*.
2. cDNA cloning and chromosomal localization of the *human and mouse* isoforms of **Ksp-cadherin**.
3. **p63** was detected in a variety of *human and mouse* tissues.

The GML system cannot link each of the gene mentions in the above sentences to just one ID. Our model can deal with these cases by modifying the constraint of Formula 2 with a larger cardinality or introducing additional formulae to determine the cardinal constraint dynamically.

Our experiment results also raise an interesting question: What causes the huge performance gap between the fine-grained and database curation evaluations? A closer look at the bottom-up approach is useful in answering this question. Several works have studied the boundary issue in entity recognition (J. Finkel *et al.*, 2005; Tsai *et al.*, 2006), and this issue was found to have a significant effect on the performance of GML. For example, consider the following sentence:

"<entity id=3083>Hepatocyte growth factor (HGF) activator</entity> is a serine protease responsible for proteolytic activation of <entity id=3082>HGF</entity> in response to tissue injury"

All of the employed gene mention recognition systems and the three open available gene mention recognition systems[3,4,5] separate the first gene mention (ID:3083) into at least one mention ("hepatocyte growth factor" or "HGF"). The incorrect boundary leads to errors in the entity mapping stage, and it could result in the extraction of an incorrect self-activation event: <entity id=3082>HGF</entity> activates <entity id=3082>HGF</entity>. An experiment conducted on the test set showed that our MLN model could achieve an F-score of 79.4% from the fine-grained IE perspective if we replaced the predicted mentions' boundaries with their corresponding overlapped gold standard boundaries. These results show that a hybrid approach combined with entity-centric boundary expansion is required before entity mapping. For instance, if we input the example sentence to a syntactic parser like Enju[6] and find that the adjacent words "Hepatocyte growth factor (HGF) activator" belong to the same noun phrase and the word "activator" is a legal suffix for a gene mention, it implies that we can expand the boundary.

The result also motivates us to reconsider the bottom-up EL approach. Are the results of entity recognition/classification a prerequisite for GML? We raise this question because, under the bottom-up approach, the entity mapping process still needs to deal with the boundary issue in order to generate more candidate identifiers, as shown by the previous example sentence. Moreover, the disambiguation process needs to look for knowledge, such as species information, surrounding the gene mention's boundary, which is usually located in the same noun phrase. It has been shown that joint learning of multiple types of linguistic structures in models can produce more consistent outputs. A feasible approach would be to treat noun phrases as potential candidate gene mentions and employ a mapping algorithm to generate identifiers for each noun phrase. Using the proposed approach to model biographical information and the dependencies between noun phrases, we can perform joint learning and inference for gene mention recognition and linking. This issue will be the major direction of our future research. For chunking parsing, there are several openly available tools, such as GENIA tagger[7], OpenNLP[8], and Lingpipe[9] package. Kang *et al.* (2011) have reported that the OpenNLP package performs noun-phrase chunking, best among the six state-of-the-art chunkers specifically for the biomedical domain. Therefore, we will use the OpenNLP

[3] http://pages.cs.wisc.edu/~bsettles/abner/

[4] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger/

[5] http://cbioc.eas.asu.edu/banner/webBasedBannerStart.html

[6] http://www-tsujii.is.s.u-tokyo.ac.jp/enju/demo.html

[7] http://www.nactem.ac.uk/GENIA/tagger/

[8] http://opennlp.apache.org/

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

package as the first step to generate candidate noun phrases, and we may consider combining more results from different chunkers through a voting strategy to further improve the chunking performance.

***Table 7. The hardest queries in the test set.***

| <query id, name, docid> |
|---|
| <#1, AIP1, 9647693> |
| <#2, TR1, 10455115> |
| <#3, PAGE1, 9651357> |
| <#4, UGT2B11, 8333863> |

Finally, we report our observations on the hardest queries under QA evaluation. Table 7 lists the queries that could not be answered correctly by any of the employed methods. After checking each error event carefully, we found that our model outputs Nil for Query #1 due to the absence of gene biographical information in the context. The gene mention of Query #2 is an abbreviation, and the query is affected by a problem similar to the ambiguous acronym problem discussed in the KBP 2009 track (McNamee, Dang, *et al*., 2009). Our model fails to output correct IDs for #3 and #4 because no distinction is made between matches of official symbols and synonyms when searching for candidate IDs. In our current work, the matches of official symbols and those of synonyms share the same predicate. We believe that appending more predicates and formulae corresponding to these two types of matches will improve our system's accuracy.

## 6. Conclusions

In this paper, we give formal definitions for EL tasks, including instance-based EL, article-wide EL, and article-wide salient EL. We then present a novel approach that employs MLN to jointly model bottom-up decisions in a specific EL task-GML. A collective formulation for instance-based GML is introduced with several useful formulae, including the dependencies among IDs, which can be used for GML disambiguation. Moreover, the benefit of predicting suitable mentions and their IDs jointly in contrast to the stage-based approach is illustrated, which selects mentions before linking IDs. Our experiments provide the first comprehensive gene mention evaluation results from three different perspectives and highlight problems that need to be addressed in the future, including the assignment of non-unique identifiers, the boundary issue, and the direction for joint entity recognition and linking.

## Acknowledgement

This research was supported in part by the National Core Facility Program for Biotechnology (Taiwan Bioinformatics Consortium of Taiwan) grant NSC-100-2319-B-010-002, the National Science Council of Taiwan grant NSC-102-2218-E-038-001, as well as the research grant TMU101-AE1-B55 of Taipei Medical University.

## Reference

Bunescu, R., & Pasca, M. (2006). Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy.

Crim, J., McDonald, R., & Pereira, F. (2005). Automatically Annotating Documents with Normalized Gene Lists. *BMC Bioinformatics*, *6*(Suppl 1), S13.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic.

Dai, H.-J., Chang, Y.-C., Tsai, R. T.-H., & Hsu, W.-L. (2011). Integration of gene normalization stages and co-reference resolution using a Markov logic network. *Bioinformatics*, *27*(18), 2586-2594.

Dai, H.-J., Lai, P.-T., & Tsai, R. T.-H. (2010). Multistage Gene Normalization and SVM-Based Ranking for Protein Interactor Extraction in Full-Text Articles. *IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS*, *7*(3), 412-420.

Dai, H.-J., Wu, C.-Y., Tsai, R. T.-H., & Hsu, W.-L. (2012). From Entity Recognition to Entity Linking: A Survey of Advanced Entity Linking Techniques. In *Proceedings of the 26th Annual Conference of the Japanese Society for Artificial Intelligence*, Yamaguchi, Japan.

Dredze, M., McNamee, P., Rao, D., Gerber, A., & Finin, T. (2010). Entity Disambiguation for Knowledge Base Population. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing.

Finkel, J., Dingare, S., Manning, C., Nissim, M., Alex, B., & Grover, C. (2005). Exploring the boundaries: gene and protein identification in biomedical text. *BMC Bioinformatics*, *6*(Suppl 1), S5.

Finkel, J. R., & Manning, C. D. (2009, June). Joint parsing and named entity recognition. In *Proceedings of NAACL 2009*.

Franzén, K., Eriksson, G., Olsson, F., Asker, L., Lidén, P., & Cöster, J. (2002). Protein names and how to find them. *International Journal of Medical Informatics*, *67*(1-3), 49-61. doi: Doi: 10.1016/s1386-5056(02)00052-7

Grosz, B. J., Weinstein, S., & Joshi, A. K. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, *21*(2), 203-225.

Hakenberg, J., Plake, C., Leaman, R., Schroeder, M., & Gonzalez, G. (2008). Inter-species normalization of gene mentions with GNAT. *Bioinformatics, 24*(16), 126-132. doi: 10.1093/bioinformatics/btn299

Hakenberg, J., Plake, C., Leaman, R., Schroeder, M., & Gonzalez, G. (2008). Inter-species normalization of gene mentions with GNAT. *Bioinformatics, 24,* i126 - 132.

Hakenberg, J. o., Royer, L., Plake, C., Strobelt, H., & Schroeder, M. (2007). Me and my friends: gene mention normalization with background knowledge. In *Proceedings of Second BioCreAtIvE Challenge Evaluation Workshop*.

Hirschman, L., Yeh, A., Blaschke, C., & Valencia, A. (2005). Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics, 6*(1).

Kang, N., van Mulligen, E. M., & Kors, J. A. (2011). Comparing and combining chunkers of biomedical text. *J Biomed Inform, 44*(2), 354-360. doi: 10.1016/j.jbi.2010.10.005

Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., & Tsujii, J. i. (2009). Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, Boulder, Colorado.

Krauthammer, M., & Nenadic, G. (2004). Term identification in the biomedical literature. *Journal of Biomedical Informatics, 37*(6), 512-526.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML).*

Lai, P.-T., Bow, Y.-Y., Huang, C.-H., Dai, H.-J., Tsai, R. T.-H., & Hsu, W.-L. (2009). Using Contextual Information to Clarify Gene Normalization Ambiguity. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IEEE IRI 2009),* Las Vegas, USA.

Li, Y., Lin, H., & Yang, Z. (2009). Incorporating rich background knowledge for gene named entity classification and recognition. *BMC Bioinformatics, 10*(1), 223.

Liu, X., Zhou, M., Wei, F., Fu, Z., & Zhou, X. (2012). Joint Inference of Named Entity Recognition and Normalization for Tweets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, Jeju, Republic of Korea.

McDonald, R., Crammer, K., & Pereira, F. (2005). Online Large-Margin Training of Dependency Parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, Michigan.

McNamee, P., & Dang, H. T. (2009a). Overview of the TAC 2009 Knowledge Base Population Track. In *Proceedings of the Second Text Analysis Conference (TAC 2009),* Gaithersburg, Maryland USA.

McNamee, P., & Dang, H. T. (2009b). Overview of the TAC 2009 Knowledge Base Population Track. In *Proceedings of the Second Text Analysis Conference (TAC 2009),* Gaithersburg, Maryland USA.

McNamee, P., Dang, H. T., Simpson, H., Schone, P., & Strassel, S. M. (2009). An Evaluation of Technologies for Knowledge Base Population. In *Proceedings of Test Analysis Conference 2009 (TAC 09)*, Gaithersburg, Maryland USA.

Mihalcea, R., & Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, Lisbon, Portugal.

Milne, D., & Witten, I. H. (2008). Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, Napa Valley, California, USA.

Morgan, A. A., Lu, Z., Wang, X., Cohen, A. M., Fluck, J., Ruch, P., . . . Hirschman, L. (2008). Overview of BioCreative II gene normalization. *Genome Biology, 9*(Suppl 2), S3.

Poon, H., & Domingos, P. (2007). *Joint inference in information extraction*.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning, 62*(*Special Issue: Multi-Relational Data Mining and Statistical Relational Learning*), 107-136.

Riedel, S. (2008). Improving the Accuracy and Efficiency of MAP Inference for Markov Logic. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI 2008)*, Helsinki, Finland.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine*, *29*(3), 93.

Tsai, R. T.-H., Wu, S.-H., Chou, W.-C., Lin, C., He, D., Hsiang, J., . . . Hsu, W.-L. (2006). Various criteria in the evaluation of biomedical named entity recognition. B*MC Bioinformatics*, *7*(92), 14.

Zhang, W., Su, J., Tan, C. L., & Wang, W. T. (2010). Entity Linking Leveraging Automatically Generated Annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, Beijing, China.

Computational Linguistics and Chinese Language Processing
Vol. 19, No. 1, March 2014, pp. 33-54

# Linking Databases using Matched Arabic Names

Tarek El-Shishtawy[*]

## Abstract

In this paper, a new hybrid algorithm that combines both token-based and character-based approaches is presented. The basic Levenshtein approach also has been extended to the token-based distance metric. The distance metric is enhanced to set the proper granularity level behavior of the algorithm. It smoothly maps a threshold of misspelling differences at the character level and the importance of token level errors in terms of token position and frequency.

Using a large Arabic dataset, the experimental results show that the proposed algorithm successfully overcomes many types of errors, such as typographical errors, omission or insertion of middle name components, omission of non-significant popular name, and different writing style character variations. When compared with other classical algorithms, using the same dataset, the proposed algorithm was found to increase the minimum success level of the best tested lower limit algorithm (Soft TFIDF) from 69% to about 80%, while achieving an upper accuracy level of 99.67%.

**Keywords:** Name Matching, Record Linkage, Data Integration, Arabic NLP, Information Retrieval.

## 1. Introduction

Information about individuals can be found in a variety of resources, such as population survey databases, national identifier databases, medical records, news articles, tax information, and educational databases. In all of these heterogeneous sources, name matching is a fundamental task for data integration that joins one or more tables to extract information referring to the same person.

Matching personal names has been used in a wide range of applications, such as record linkage or integration, database hardening, removing or cleaning up duplicated records, and searching the Web. Unfortunately, the name may not be known exactly, may be misspelled, or may have spelling variations. Therefore, in these applications, the general word matching

[*] Prof. Ass. of Computer Engineering, Faculty of Computers and Information, Benha University
E-mail: t.shishtawy@ictp.edu.eg

techniques are insufficient, and optimized techniques have been developed to cope with matching multiple variations of the same personal name.

There are efficient and well-established algorithms that deal with spelling errors, variants for strings, and name matching at a character level. For relatively short names that contain similar yet orthographically distinct tokens, character-based measures are preferable because they can estimate the difference between the strings with higher resolution (Bilenko *et al*., 2003a; Bilenko & Mooney, 2003). In languages where names have very close typographic structure, such as Arabic, character level similarity is not enough to produce high precision matching.

Unfortunately, spelling errors are not the only source of name mismatching. People may report their names inconsistently by removing or inserting additional name tokens, adding initial titles, or writing different punctuation marks and whitespaces. In all of these cases, bag-of-words methods are better suited to the matching problem, since they are more flexible at word level. In addition, token-based approaches are not able to capture the degree of similarity between individual tokens with minor variations in characters (Bilenko *et al*., 2003b).

Experimental results show that hybrid techniques, which take word frequency as well as character-based word similarities into account, increase matching. A first attempt in this direction was introduced by Cohen *et al*. (Cohen *et al*., 2003), in the form of a measure called Soft-TFIDF, which extends the Jaro-Winkler method to combine both the frequency weight of words in a cosine similarity measure and a CLOSE measure at the character level. The soft TFIDF algorithm works as follows: for each token $A_i$ in the first name, find a word $B_j$ in the second one that maximizes the similarity function. Moreau *et al*. (Moreau *et al*., 2008) showed that this may lead to double matching of words and proposed a generic model to enhance the soft TFIDF. Camacho *et al*. (Camacho *et al*., 2008) used a cost function that basically depends on matching all pairs of tokens and summing all edit distances at character level. The distance metric was modified by a frequency measure of the tokens. They also used a permutation factor – Monge-Elkan concept (Monge & Elkan, 1996) – to allow a non-ordered sequence of word tokens to be matched.

In this work, we propose a hybrid sequential algorithm that combines the advantages of token level and character level approaches to improve the name matching quality. The hybrid algorithm is optimized for matching Arabic names. As we will discuss in Section (3), Arabic names have a restricted writing order and close typographic pattern and are subject to middle token omission and omission of common name tokens, even if occurring at the beginning of names. Due to these characteristics, existing algorithms cannot be applied directly to matching Arabic names. Character-based hybrid algorithms may fail due the close typographic pattern, ignoring the sequence order of tokens. Also, allowing permutation of tokens conflicts with the restricted sequence of writing names. To improve the matching efficiency, most hybrid techniques include weights for token frequency but do not give the same attention to the relative position where the mismatch occurs.

The proposed hybrid algorithm is an extension to the Levenshtein algorithm, with computing 'edit distances' at token level instead of character level in the basic algorithm. The sequential nature of the algorithm keeps the ordering importance of name tokens. The two names to be matched are considered as two bags of tokens, and the algorithm computes the cost of transforming one bag into the other. While the basic Levenshtein algorithm assigns a unity cost to all edit operations, the current algorithm assigns weights that reflect the importance of each edit operation. When matching a pair of tokens, the importance of the edit operation is determined by the relative position of the tokens in names, their frequency measure, and their character level partial similarity.

The remaining parts of this paper are organized as follows. In Section 2, we present some basic techniques for name matching. After that, Section 3 briefly discusses the characteristics of the Arabic naming system considered in our work. The proposed algorithm is described in Section 4. The results of experimental comparisons are discussed in Section 5. Finally, conclusions are discussed in Section 6.

## 2. Matching Algorithms for Names

Name matching is the process of determining whether two name strings are instances of the same name. Multiple methods have been developed for matching names, which reflects the large number of errors or transformations that may occur in real-life data (Elmagarmid *et al*., 2007). The basic goal of all techniques is to match names (or strings) that are not necessarily required to be identical. Instead of an exact match, a normalized similarity measure usually is calculated to have a value between 1.0 (when the two names are identical) and 0.0 (when the two names are totally different). There are several well-known methods for estimating similarity between strings, which can be roughly separated into two groups: character-based techniques and token-based techniques.

The Levenshtein algorithm and its variants are character-based matching techniques based on edit distance metrics, and the Levenshtein edit distance is defined originally for matching two strings of arbitrary lengths. It counts the minimum differences between strings in terms of the number of insertions, deletions, or substitutions required to transform one string into the other. A score of zero represents a perfect match.

The basic Levenshtein method has been extended in many directions (Hall & Dowling, 1980), for example, having an extension to consider reversals of order (transposition of characters) directly in the edit distance operation. Another direction of generalization is to allow different weights at character level. The weights for replacing characters can depend on keyboard layout or phonetic similarities (Snae, 2007). In other research (Bilenko *et al*., 2003b), a distance function is produced by a distance function learner and the weights are learned from a training data set to have a combined record-level similarity metric for all fields.

The affine gap distance metric (Waterman *et al*., 1975) offers a smaller cost for gap mismatch; hence, it is more suitable for matching names that are truncated or shortened. Smith and Waterman (1981) described an extension of edit distance and affine gap distance to find the optimal local alignment at the character level. Regions of gaps and mismatches are assigned lower costs. Jaro (1989) introduced a string comparison metric that is dependent on both the number of common characters and the number of non-matching transpositions in the two strings.

Token-based approaches are motivated by the fact that most of the differences between similar named entities often arise because of abbreviations or whole word insertions and deletions. Hence, token models should produce a more sensitive similarity estimate than character-based approaches. Also, experimental results show that including token frequency as a parameter in matching algorithms leads to a significant improvement in matching accuracy. Jaccard-vector space cosine similarity is an example of difference measures that operate on tokens, treating a string as a “bag of words”. In these approaches, the two string names to be compared are divided into sets of words (or tokens) before a similarity metric is considered over these sets.

The Jaccard similarity between the word sets A and B is defined as the size of the intersection divided by the size of the union of the sample sets: J (A,B) = |A ∩ B| / |A ∪ B|. The algorithm has been extended to compare bi-grams (paired characters of two string), tri-grams, or n-grams. Strings can be "padded" (Keskustalo *et al*., 2003) by adding special characters at the beginning and end of strings, Padded n-grams will result in a larger similarity measure for strings that have the same beginning and end but errors in the middle.

TFIDF, or cosine similarity, is another measure that is widely used in the information retrieval community. The basic TFIDF makes uses of the frequency of terms in the entire collections of documents and the inverse frequency of a specific term in a document. The TFIDF weighting method is often used in the Vector Space Model together with Cosine Similarity to determine the similarity between two documents. Similarity between database strings, or between a database string and a query string, is computed via the cosine similarity (inner product) of the corresponding weight vectors, essentially taking the weights of the common tokens into account. The TFIDF similarity of two word sets A and B can be defined as:

$$\mathrm{TFIDF}(A,B) = \sum_{w \in A \cap B} V(w,A).V(w,B) \tag{1}$$

where V is a weight vector that measures the normalized TFIDF of word w. Like Jaccard, the TFIDF scheme depends on common terms, but the terms are weighted; these weights are larger for words w that are rare in the collection of strings from which A and B are drawn. The basic TFIDF does not account for misspelling mistakes in words. Cohen *et al*. (2003) proposed a soft TFIDF with a heuristic that accounts for certain kinds of typographical errors.

Soft TFIDF is one approach that combines both string-based and token-based distances. In this approach, similarity is affected not only by the tokens w that appear in the sets A and B, but also for those "similar" tokens in A that appear in B.

$$\mathrm{softTFIDF}(A,B) = \sum_{w \in close(\theta,A,B)} V(w,A).V(w,B)D(w,B) \qquad (2)$$

Here, D is the character-based distance of the word w, such that it is greater than a threshold θ. The new set close allows one to integrate a token-based distance and the statistics of a particular corpus in the similarity evaluation of a particular word.

Both TFIDF and Soft TFIDF are insensitive to the location of words, thus allowing natural word moves and swaps (*e.g*., "John Smith" is equivalent to "Smith, John"). Although this is useful in many naming systems, it does not fit the Arabic naming system, which is characterized by restricted component order. Camacho *et al*. (2008) proposed a similar metric that combines both the frequency of words and the edit-based distances of each word pair of the two names. Also, strings may be phonetically similar even if they are not similar at the character or token level. Soundex (Holmes & McCabe, 2002), Phonex (Gadd, 1990), and Phonix (Gadd, 1990) are examples of phonetic-based techniques that convert the name into a sequence of codes that represent how the name is spoken. Phonetic representation of the names is used either for exact or approximate match.

When considering contextual information stored with names (such as address, mail, and other details) to increase the likelihood of a match, the problem is called data or record linkage (Xiao *et al*., 2011). Many techniques have been proposed for record linkage, where not only are pairs of name strings matched, but also many other matching features (Monge & Elkan, 1996). Winkler (2002) demonstrated how machine-learning methods could be applied in record linkage situations where training data are available. Name is time-independent information; therefore, even in feature-based approaches, having an effective name matching is crucial (Winkler, 2006). This work considers only name matching without taking any contextual information into account.

## 3. Characteristics of Arabic Name Variations

Exact string matching of personal names is problematic for all languages because names are often queried in a different way than they were entered. The proposed algorithm deals with the following problems concerning Arabic names.

## 3.1 Very Close Typographic Structure

Spelling errors normally occur during data entry. This may be due to typographical errors, cognitive errors, or phonetic errors. Whatever the reason for the error, the source and target names are considered strings differing at the character level. According to Jurafsky and Martin (Jurafsky *et al*., 2002), this type of error can be categorized as insertion, deletion or omission, substitution, or transposition.

There are efficient and well-established algorithms that deal with spelling errors variants for a string. When data is represented by relatively short strings that contain similar yet orthographically distinct tokens, character-based measures are preferable since they can estimate the difference between the strings with higher resolution.

The reason that misspelling errors are particularly difficult in Arabic names is the close typographical structure of names. For example, inserting the character "و" to the name "محمد," yields a correct name "محمود". Also, substituting the character "أ" with "م" in "دمحم," gives a correct name "أحمد". If the Levenshtein algorithm is used for matching two names with a length of 20 characters of each name, a single edit distance will show 95% matching similarity of the two names, while they are two different persons. The problem is how to know if the name is written incorrectly or refers to another person (*e.g*., his brother), especially when searching family databases.

## 3.2 Omission of Name Components

While it is common to have one first, one or more middle, and a surname name for writing a personal name, several variations exist in real free form names. The same problem exists in many other languages, and it has been reported by Borgman and Siegfried (Borgman & Siegfried, 1999) that there are no legal regulations of what constitutes a name. The source of the ambiguity, in many cases, is people themselves as they report their names differently depending upon the organization they are contacting. Examining different Arabic databases shows that name omission is a serious problem that should be handled efficiently in any Arabic name matching algorithm. Name omission is related to both position and frequency as follows.

### 3.2.1 Name Order

*Persons tend to write their names in a restricted correct order. They may omit one or more tokens, while still keeping the correct order.*

Examining different writing styles of Arabic names shows that transposition errors occur rarely. Therefore, one wants “Hamed Mohamed Fawzy Ibrahim” to match with “Hamed Mohamed Ibrahim” but not with “Hamed Fawzy Mohamed Ibrahim”. The built in sequential nature of the proposed algorithm assigns one edit distance to 'omission' and 'two edit distances' to transposition.

### 3.2.2 Position of Omitted Name

*Persons tend to omit one or more middle names, while fewer name omissions typically occur at the beginning or at the end of names.*

The analysis shows that a person is keen on writing his first and surname carefully. This raises the position importance of the name variations. For example, one wants “Hamed Mohamed Fawzy Ibrahim” to match with “Hamed Mohamed Ibrahim” but not with “Mohamed Fawzy Ibrahim”. To realize the position relation, the proposed algorithm gives less importance to name omission – or insertion – occurring in the middle of the name, and more importance to first and last names.

### 3.2.3 Frequency of Omitted Name

*Persons tend to omit non-significant components of their names, i.e., omission is likely to occur with common names.*

Results of analyzing a sample of 7140 Egyptian full names show that nearly 30% of all name components lie within a set of only 9 common names, as shown in Table (1). In the proposed algorithm, less importance is given to a common name omission or insertion. For example, one wants “Hamed Mohamed Fawzy Ibrahim” to prefer the match with “Hamed Fawzy Ibrahim” over “Hamed Mohamed Ibrahim,” because 'Mohamed' is not as indicative a name as ' Fawzy'.

***Table 1. Arabic Common Name Frequency***

| Arabic Name | English name | TF |
|---|---|---|
| محمد | Mohamed | 11.38% |
| احمد | Ahmed | 5.98% |
| محمود | Mahmoud | 2.39% |
| على | Ali | 2.28% |
| ابراهيم | Ibrahim | 2.07% |
| حسن | Hassan | 1.84% |
| السيد | Alsayed | 1.54% |
| مصطفى | Mostafa | 1.33% |
| حسين | Hossien | 0.87% |
| Total percentage of top common Arabic name tokens | | 29.7% |

Common names have another impact when searching the Web for famous persons. When searching for the former president of Egypt, many people do not know that his first name is 'Mohamed,' and search the web only for 'Hossny Mubark". The search engine should be smart enough to return also matches with his full name as top hits, because 'Mohamed' is a common name and is expected to be omitted. This is different from returning 'Gamal Hossny Mubark' – his son – since 'Gamal' is not a common name.

The previous discussion shows that the frequency distributions of name values can be used to improve the quality of name matching. They can be calculated either from the data set containing the names to be matched or from a more complete population-based database, like a telephone directory or an electoral roll.

## 3.3 Writing Styles Character Variations

One important component of the proposed work is name standardization. Standardization eliminates writing style character variations; hence, it makes the data comparable and more usable. To produce a uniform representation, the algorithm runs SQL script to replace various spellings of words with a single spelling. For instance, different prefixes, spacing, punctuations, and character variations are replaced with a single standardized spelling.

A name standardization (or character variation elimination) module is common module in name matching algorithms (PAtman & Thompson, 2003). In the current work, the standardization concept is optimized for Arabic names. For instance, the module trims certain prefixes such as (د. أ.د. م. دكتـــور, د/ ، الســـــيد/), replaces multiple blanks with a single blank, replaces the characters (أ، إ ، آ with ا), and replaces the ending character (ي) with (ى). There are some cases where the Arabic name component is composed of two tokens. For example, a prefix name component (عبـــد) and a postfix name component (الـــدين) cannot be standalone names. There is no standard style for writing composite names, as it is not always necessary to have a distance space between them. To standardize composite names, either leading or trailing spaces are removed, whenever a pre/post tokens are detected. Name style standardization is an inexpensive step, but it improves the overall performance for name matching.

## 4. The Proposed Algorithm

We started with the Levenshtein edit distance similarity metric and extended it to handle name matching at a token level. The sequential nature of the Levenshtein method ensures that the sequential name order of tokens is considered. Following the variant of Needleman-Wunch (gap cost), the current algorithm replaces the fixed unity cost of the simple Levenshtein form with a cost function that is dependent on frequency and position of tokens to be matched.

Specifically, the implementation of the proposed algorithm applies three modifications to the basic Levenshtein distance metric. The first modification is the application of the same dynamic programming technique at the token level instead of the character level in basic Levenshtein. For example, the distance between the two names a = ('Mohamed,' 'Ahmed,' 'Hassan,' 'Ali') and b = ('Mohamed,' 'Hassan,' 'Ali,' 'Ibrahim') is two. This is because (a) requires two edit operations (deletion of the token 'Ahmed', and insertion of the token 'Ibrahim' at the end of a).

The second modification is the mapping of the frequency and position importance of name tokens – discussed in Sections 3.2 and 3.3 – with a cost function C, instead of assigning fixed unity cost for all edit operations. The role of the cost function C is to lighten (or strengthen) the effect of token mismatch according to word position and frequency.

The third modification is the implementation of partial matching of individual token pairs at the character level. This fine grained level ensures that pairs with slightly different misspellings are not ignored.

For two tokens $(\mathrm{a_k}, \mathrm{b_l})$, where $1 \le \mathrm{l} \le \mathrm{L}$ and $1 \le \mathrm{k} \le \mathrm{K}$

$$H(k,l) = min \begin{cases} \mathrm{H}[\mathrm{k}-1,\mathrm{l}] + C_{k,l} \\ \mathrm{H}[\mathrm{k},\mathrm{l}-1] + C_{k,l} \\ \mathrm{H}[\mathrm{k}-1,\mathrm{l}-1] + \mathrm{TokenCost}(a_k, b_l)C_{k,l} \end{cases} \tag{3}$$

where $\mathrm{C_{k,l}}$ is given by

$$C_{k,l} = P_{k,l}\ F_{k,l} \tag{4}$$

$\mathrm{P_{k,l}}\ \mathrm{F_{k,l}}$ are the position and frequency costs defined in Sections (4-2) and (4-3), respectively.

TokenCost is the token-pair similarity cost at a character level. The final similarity percentage between the two name strings then is given by:

$$\mathrm{sim}[\mathrm{a},\mathrm{b}] = 1 - \frac{\mathrm{H}[\mathrm{K},\mathrm{L}]}{\mathrm{max}(\mathrm{K},\mathrm{L})} \tag{5}$$

## 4.1 Token Pair Mismatch Cost

The 'TokenCost' measure is the cost of partial match between tokens a[k] and b[l], and it captures word spelling errors at the character level. Pairs of tokens that are not necessarily identical are also considered in the edit operation but with a cost that depends on their similarity. The proposed 'TokenCost' measure combines the token level edit operations with approximate token matches. The concept is similar to the 'close' function in soft TFIDF. In our implementation, the 'TokenCost' function has a value that ranges from zero (for exact token match, thus having 0 required edit actions), to 1 (for completely mismatch, thus having 1 complete edit action). When similarity is below a certain threshold, the pairs are considered dissimilar and the distance function is set to one. It is computed with Levenshtein edit distance, and clipped at a threshold value θ. The threshold limit is implemented with the threshold rule:

$$\begin{aligned}&\mathrm{TokenCost}(a_k, b_l) = \mathrm{Levenshtein}(a_k, b_l)\\ &\mathrm{IF\ Levenshtein}(a_k, b_l) \geq \theta \mathrm{\ then\ TokenCost}(a_k, b_l) = 1\end{aligned} \tag{6}$$

For two names a and b, the current algorithm defines a Levenshtein cost of the two token words $a_k \in a$ and $b_l \in b$ as $0 \leq \mathrm{Levenshtein}(a_k, b_l) \leq 1$. The distance cost ranges from zero for a complete match to one for a complete mismatch. The distance depends on Levenshtein edit metric. The basic Levenshtein algorithm is a character-based approach, which takes two strings, $a_k$ and $b_l$ of lengths m and n characters, and returns the Levenshtein distance between them. For all i and j, d[i,j] will hold the Levenshtein distance between the first i characters of $a_k$ and the first j characters of $b_l$. The elements of d[i,j] are computed according to the following:

$$\begin{aligned}&\mathrm{If}\ a_k[i] = b_l[j], \mathrm{then}\ d[i,j] = d[i-1, j-1]\\ &\mathit{Else}\ d[i,j] = \mathit{min}\begin{cases} d[i-1, j] + 1 & \mathit{//a\ deletion}\\ d[i, j-1] + 1 & \mathit{//an\ insertion}\\ d[i-1, j-1] + 1 & \mathit{//a\ Substitution}\end{cases}\end{aligned} \tag{7}$$

The similarity measure then is given by:

$$\mathrm{Levenshtein}[a_k, b_l] = \frac{d[m,n]}{\max(M,N)} \tag{8}$$

## 4.2 Position Mismatch Cost of Tokens

As in Western naming systems, an Arabic name’s token order is very important. Family name usually appears as the last token of the name. Therefore, any successful name matching algorithm should allow for gaps of unmatched characters (*e.g.*, Smith-Waterman algorithm) and the problems of out of order of tokens (*e.g.* Monge-Elkan method). The proposed algorithm satisfies both constraints with more flexibility for adjusting the relative importance of token position.

Instead of using the number of edit operations as a distance metric, the proposed algorithm uses the cost of the edit operations required to transform one string to another. When matching complete names, inserting or deleting a token (name) at the beginning (first name) or at the end (family name) of a complete name has a different cost from doing the same edit actions for middle names. We call this position cost, which is implemented using a position weight cost $0 \leq P \leq 1$.

$P_{k,l}$ is the position weight for a matching token k with a token j. In general, the proposed position cost is flexible and can have different values at each token position. In our implementation, persons are keen to write their first and last names. For this reason, more importance is given when matching first and last names. Position weight is assigned a complete unity edit cost when matching either the two first or the two last tokens in both strings.

In our implementation, for two tokens $(a_k, b_l)$, where 1 ≤ l ≤ L and 1 ≤ k ≤ K, the position cost rule is given by:

$$P_{k,l} = \begin{cases} 1, & (k=1 \text{ and } l=1) \\ & \text{or}(k=K \text{ and } l=L) \\ \beta & \text{Otherwise} \end{cases} \tag{9}$$

The position rule is used to initialize the zero rows and columns, as shown in Table 1. Note that this initialization is different from Smith and Waterman (1981), which assigns zeroes to all zero rows and columns. Table (2) illustrates an example of computing H(l,k), defined in Equation (4), when considering only position weights to match two strings a [L]= (W1, W2, W3, W4, W5) and name b[K]= (W1, W3, W4, W6).

***Table 2. Effect of position weight***

| | | W1 | W3 | W4 | W6 |
|---|---|---|---|---|---|
| | 0 | 1 | 1+β | 1+2β | 2+2β |
| W1 | 1 | 0 | β | 2β | 3β |
| W2 | 1+β | β | β | 2β | 3β |
| W3 | 1+2β | 2β | β | 2β | 3β |
| W4 | 1+3β | 3β | 2β | β | 2β |
| W5 | 2+3β | 4β | 3β | 2β | 1+β |

As given in Equation (9), the position weight has either a value of 1 or β. In Table (2), and according to Equation (9), the step increase of position weight (in zero rows and columns) is β, except for the first and last cells, which have a value of one. The remaining cells are computed based on Equation (4) while considering only position weight. For example, the entry at a cell (W2, W3) is the minimum of its up, left, and left-up cells plus β, which gives H(W2,W3)=β. Another example, H(W3.W3) is β, since the two words are matched and the IF-part of Equation (4) is applied.

The total distance cost then is H(W5,W6) divided by the maximum length of the two tokens, which is (1+ β) /5, Note that, in the original edit distance algorithm, a complete edit distance (β =1) is assigned to each position. By varying the value of (β≤1), one can control the position importance of the token.

### 4.3 Frequent Name Mismatch Cost

Motivated by the assumption that 'persons tend to omit their common names,' we allow the cost of edit distances for a common name to have a different cost than a rare name. This concept is called a term frequency weight $0 \le F \le 1$, For two tokens $(a_k, b_l)$, where $1 \le l \le L$ and $1 \le k \le K$, the term frequency is given by:

$$F_{k,l} = 1 - \alpha \frac{TF(a_k) TF(b_l)}{MTF^2} \tag{10}$$

where $TF(a_k)$ is the Term Frequency of token $a_k$, $TF(b_l)$ is the Term Frequency of token $b_l$, MTF is the maximum term frequency of names, and α is a frequency weighting factor. To allow the algorithm to have a maximum frequency effect, α is set to 1. In this case, the common name frequency cost and the overall cost of the edit operation are nearly 0.

## 5. Experimental Results

### 5.1 sets

We used two types of datasets: the base set and test sets. The base dataset contains a sample of 7140 names extracted from MIS of Egyptian university staff members. To make the base dataset usable, all names were standardized to eliminate writing style character variations, as explained in Section 3.3. The extracted sample contains two fields: 1) a Base name IDentifier B_ID, and the Base Name (BName).

The Arabic names dataset characteristics are shown in Table (3) and can be downloaded from http://www.scribd.com/doc/143493637/Arabic-Name-Dataset. The frequency distribution of the number of tokens in the sample indicates that 97.4% of the Arabic names were written in 3 to 5 tokens. The preliminary analysis of the sample shows that the sample contains 30 duplicated names (a percentage of 0.41%). Also, as shown in Table (3), the duplication of names is only 0.76% (actually 0.35% when subtracting full name duplications) when the Arabic name is written in four tokens.

The experiment used six different test datasets, with each containing 300 names extracted randomly from the base dataset. Each test set was subjected to a noise to induce one of the following errors: 1) deletion of random single character, 2) deletion of random two characters, 3) omitting the first token, 4) omitting the second token, 5) omitting the third token, or 6) omitting both the second and third tokens. Each test set had one type of distortion, but all test sets had a similar field structure: 1) Test name Identifier (T_ID), 2) the distorted name (DName), and 3) a Reference to the original base name Ref_B_ID. Then, we had six test sets for which the true match status was known and each carrying one type of errors.

***Table 3. Characteristics of the Arabic name dataset***

<table>
<tr><th>No. of Words</th><th>Frequency</th><th colspan="5">Duplication</th></tr>
<tr><td>1</td><td>0</td><td>88.34%</td><td rowspan="2">35.64%</td><td rowspan="3">5.74%</td><td rowspan="4">0.76%</td><td rowspan="7">0.41%</td></tr>
<tr><td>2</td><td>0.1%</td><td rowspan="6"></td></tr>
<tr><td>3</td><td>14.4%</td><td rowspan="5"></td></tr>
<tr><td>4</td><td>62.6%</td><td rowspan="4"></td></tr>
<tr><td>5</td><td>20.4%</td><td rowspan="3"></td></tr>
<tr><td>6</td><td>2.2%</td></tr>
<tr><td>7</td><td>0.2%</td></tr>
</table>

## 5.2 Experimental Methodology

The purpose of all experiments was to measure the degree to which the proposed algorithm could overcome each type of inserted noise. The 300 distorted names of each test set were matched against the original 7140 base set. Since we are interested in automating the matching process, we defined 'success' as a direct measure of the algorithm performance. The testing environment was adopted to accept only the top-scoring name from the tested algorithm as a match. The match was counted as true if it corresponded to its original name in the base set. The success percentage was calculated as the count of true matches divided by the test set size.

For a single distorted name (DNname, Ref_B_ID), the test methodology consults the tested algorithm (the proposed algorithm or other algorithms), to compute the similarity against the base set. The test environment keeps the running maximum similarity and the Base name ID (Sim, B_ID) as a candidate matched name. This match is true when the final maximum similarity name is the same as the original base name. The algorithm is given as follows.

```
Algorithm Success-Match-Percentage
Input :         Specific values of $\theta, \alpha$ and $\beta$
            Specific Distorted Data Set $DDS_i$ (DNname, Ref_B_ID) ,   $1 \leq i \leq 300$
            Base Data Set $BDS_j$ (BName, B_ID), ,   $1 \leq j \leq 7140$
Begin
   True Match Count = 0;
   For (I = 1 to 300) {
          Max Running score = 0;
          Ref record = 0;
          For (j=1 to 7140) {
                    Apply Equation (5) to get Sim (DName(i), BName(j)
                              If Sim> Running max Score {
                    Set Max Running Score = Sim;
                    Set Ref Record = B_ID (j); }
             } End loop j
        If    Ref record = Ref_B_ID (i) {        /* success */
             Increment True Match Count by 1 ; }

  } End loop i
 Success percentage = True Match Count / 300;
End;
```

In summary, we used artificially generated files, with each carrying one type of error. Although the generated data sets do not approximate the types of errors that occur in real data, they are very useful in 1) analyzing the effect of different parameters on each type of error and 2) determining the upper and lower success limits of each algorithm.

## 5.3 Results

We ran our experiments on data sets, with each carrying only one type of error. Results of the proposed algorithm are summarized in Tables 4 to 9. Each cell entry is the percentage of success, which is calculated as the number of true matches of the test set names divided by the total number of names in the set. To illustrate the role of the position weight and individual token match threshold in evaluating a name matching, the experiment was repeated for different values of β and θ.

***Table 4. One character omission success percentages***

| *θ* | β | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 0.1 | 0.3 | 0.5 | 07 | 1 |
| 0 | 50.33% | 70.33% | 91.00% | 93.67% | 94.67% | 94.00% |
| 0.1 | 50.33% | 70.33% | 91.00% | 93.67% | 95.67% | 94.00% |
| 0.3 | 82.33% | 91.00% | 95.33% | 95.33% | 97.67% | 95.33% |
| 0.5 | 87.67% | 97.00% | 97.33% | 99.00% | 99.00% | 99.33% |
| 0.7 | 88.00% | 97.33% | 98.67% | 98.67% | 98.00% | 98.33% |
| 1.0 | 87.33% | 97.00% | 98.67% | 98.67% | 98.00% | 98.00% |

***Table 5. Two characters omission success percentages***

| *θ* | β | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 0.1 | 0.3 | 0.5 | 07 | 1 |
| 0 | 29.33% | 39.00% | 64.00% | 70.67% | 71.33% | 72.00% |
| 0.1 | 29.33% | 39.00% | 64.00% | 79.67% | 90.67% | 82.00% |
| 0.3 | 67.33% | 78.33% | 88.67% | 90.33% | 94.67% | 91.00% |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0.5 | 79.67% | 92.00% | 96.00% | 96.33% | 97.33% | 96.67% |
| 0.7 | 78.67% | 94.33% | 96.67% | 96.67% | 96.33% | 95.33% |
| 1.0 | 78.67% | 94.33% | 96.67% | 96.67% | 95.67% | 95.00% |

***Table 6. First token omission success percentages***

| *θ* | β | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 0.1 | 0.3 | 0.5 | 07 | 1 |
| 0 | 2.33% | 34.00% | 74.33% | 82.33% | 87.00% | 89.00% |
| 0.1 | 2.33% | 34.00% | 74.33% | 82.33% | 87.00% | 89.00% |
| 0.3 | 2.67% | 28.67% | 66.33% | 76.67% | 85.33% | 88.67% |
| 0.5 | 4.00% | 16.33% | 53.00% | 67.33% | 81.33% | 84.67% |
| 0.7 | 4.00% | 10.00% | 41.33% | 63.67% | 78.67% | 82.00% |
| 1.0 | 4.33% | 8.33% | 37.00% | 62.33% | 76.67% | 79.33% |

***Table 7. Second token omission success percentages***

| *θ* | β | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 0.1 | 0.3 | 0.5 | 07 | 1 |
| 0 | 92.33% | 99.33% | 99.33% | 99.67% | 99.67% | 99.00% |
| 0.1 | 92.33% | 99.33% | 99.33% | 99.67% | 99.67% | 99.00% |
| 0.3 | 92.33% | 99.00% | 98.67% | 98.00% | 97.33% | 96.67% |
| 0.5 | 92.33% | 98.67% | 98.33% | 97.33% | 96.00% | 93.33% |
| 0.7 | 92.33% | 98.00% | 97.67% | 97.00% | 95.67% | 90.00% |
| 1.0 | 92.33% | 97.33% | 97.00% | 96.67% | 95.33% | 87.67% |

***Table 8. Third token omission***

| *θ* | β | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 0.1 | 0.3 | 0.5 | 07 | 1 |
| 0 | 77.00% | 83.67% | 91.00% | 94.33% | 95.00% | 71.00% |
| 0.1 | 77.00% | 83.67% | 91.00% | 94.33% | 95.00% | 71.00% |
| 0.3 | 77.00% | 83.33% | 89.00% | 90.67% | 93.67% | 65.67% |
| 0.5 | 77.00% | 83.33% | 85.67% | 89.33% | 93.00% | 60.67% |
| 0.7 | 77.00% | 83.33% | 84.67% | 89.00% | 92.67% | 55.00% |
| 1.0 | 77.00% | 82.33% | 83.00% | 88.00% | 92.00% | 51.67% |

**Table 9. Second and third tokens omission success percentages**

| $\theta$ | β | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 0.1 | 0.3 | 0.5 | 07 | 1 |
| 0 | 76.67% | 76.33% | 79.67% | 79.67% | 80.33% | 71.00% |
| 0.1 | 76.67% | 76.33% | 79.67% | 79.67% | 80.33% | 71.00% |
| 0.3 | 76.67% | 76.00% | 78.00% | 78.00% | 77.00% | 65.67% |
| 0.5 | 76.67% | 76.00% | 78.00% | 78.00% | 74.00% | 60.67% |
| 0.7 | 76.67% | 76.00% | 78.00% | 78.00% | 73.67% | 55.00% |
| 1.0 | 76.67% | 75.67% | 77.33% | 77.33% | 72.00% | 51.67% |

In general, the results show that the algorithm succeeds in overcoming single character and second token omissions with nearly 100% accuracy when properly setting the parameter values. Names with two character omission are matched successfully with 97.33% accuracy. Next is the third token omission with a success percentage of 95.00%. Table (6) shows that the first token omission is a serious error, and the algorithm succeeds only with 89% accuracy in matching correct names. The worst accuracy of 79.67% is achieved when a person omits both his/her second and third names.

### 5.3.1 Effect of Threshold and Position Weights

The results show that there is a conflicting need for the value of θ required for obtaining optimum results for all types of errors. Character omission errors require a moderate value (θ = 0.5), while overcoming token omission requires lower values of θ ranging from 0 to 0.1. It is clear that, for lower thresholds, the similarity of token cost at character level (the third term of the minimum relation of Equation 3), always dominates, and the system behavior is mainly a character level operation. On the other hand, setting the threshold θ = 1, a unity edit cost is assumed and the algorithm neglects all character variations at the token level.

The position weight β required to obtain best results has a fixed pattern for both characters and token omission errors. Better results usually are found when β ranges from 0.5 to 0.7. The only exception is the first token omission error, which needs higher values of β. To explain the effect of β values, let us return to Equation (3). If β has a maximum value of 1, no positional information will be used, since a unity edit cost is assumed whatever the position is and the algorithm behavior will give equal importance to all token variations, which corresponds to a static string distance. Also, for very low values of β, the algorithm neglects nearly all token level variations other than first and last tokens.

### 5.3.2 Comparing Results with other Algorithms

We now present an experiment comparing our hybrid weighting technique (run at α =1, β=0.7,θ =0.1) with different state-of-the-art systems using the same Arabic data sets and the same experimental methodology. The tested algorithms are: Levenshtein, Monge-Elkan, Jaro-Winkler (JW), and Soft-TFIDF (run at JW, θ =0.9). The Java open-source toolkit of tested algorithms is available at http://secondstring.sourceforge.net/) (Cohen *et al.*, 2003).

When considering only character misspelling errors, the basic Levenshtein algorithm did the best, followed by Jaro-Winkler with a success of 95%. Both the proposed algorithm and Soft TFIDF had the same average success rate of 93% for character level errors, while the Monge-Elkan algorithm returned the worst accuracy result, where nearly 45% of the distorted names were matched incorrectly.

For token omission errors, as shown in Table 10, our proposed algorithm generally seems the best when considering either the success boundary or individual token's errors. It had a better success range (from 80.3% to 99.7%) than Soft TFIDF (from 69.0% to 94.3%). Part of this better performance may be due to missing prior Arabic frequency knowledge for the Soft TFIDF algorithm. The next best performance for the tested algorithms is the Monge-Elkan method, which had a success boundary ranging from 23.0% to 89.3%. The worst result comes from Jaro-Winkler, with success ranging from 8.3% to 72.3%.

***Table 10. Comparison of success percentages of the proposed algorithm with existing matching algorithms using Arabic dataset.***

| Omission Error Type | Proposed Average β=0.7 θ =0.1 | Basic Lev. | Monge Elkan | Jaro Winkler (JW) | Soft TFIDF JW θ =0.9 |
|---|---|---|---|---|---|
| One character | 95.7% | 100% | 66.7% | 96.0% | 95.7% |
| Two characters | 90. 7% | 100% | 44.3% | 94.0% | 90.3% |
| First token | 87.0% | 90.7% | 89.3% | 72.3% | 86.0% |
| Second token | 99.7% | 67.7% | 40.3% | 68.7% | 94.3% |
| Third token | 95.0% | 65.3% | 35.0% | 51.3% | 91.3% |
| Second & third | 80.3% | 16.0% | 23.0% | 8.3% | 69.0% |

It is important to note that the actual success percentage of all presented algorithms will be at a point between their upper and lower boundary limits in real data sets. The actual success operation of any algorithm depends also on the mixture amount of different types of errors in the real sets.

### 5.3.3 Practical Implementation of the Algorithm

The proposed algorithm was built as a tool for integrating cluttered and heterogeneous databases in universities as a part of activities of the Egyptian Universities Portals Project EUP. The project was funded by Ministry of Higher Education MOHE in Egypt. The official data of staff members are stored in SQL server databases – across universities – and include many fields, such as primary key fields, staff national ID, staff name, and address. Also, there exist many simple databases (in many cases only Excel sheets), that hold other activities within each university, such as Training File Records. Training records include staff name, course title, completion date, and other attributes. For each new course, new records are added to the file describing attendance with repeated staff member names. Therefore, the only available field for integrating both databases to get all courses for a staff member is matching the names in both databases.

The proposed algorithm was adopted to copy primary key fields from staff MIS table as foreign key fields in training database tables when names in both tables are matched. In Benha University, the staff table holds data of nearly 3500 staff members, while the number of records in the training database was 2200. When trying to link both databases using exact name matching, only 14% of names stored in the training database were identified. In implementing the proposed algorithm, the success in linking both tables reached 98.5% without any manual intervention. It is important to note here that the success factor greatly exceeds the minimum success level described previously through testing experiments. This is because the implemented heavy distortion scenarios were the worst cases that any matching algorithm should overcome. For example, real names to be matched cannot all have two missed tokens. Nevertheless, this high linking success level cannot be generalized since it is strongly dependent on the nature of errors existing in the names to be matched. Therefore, we highlight the importance of setting referenced criterion based on error types when comparing the success percentage of matching algorithms.

## 6. Conclusions

In this work, the basic character-based Levenshtein approach has been extended to a token-based distance metric. The algorithm is enhanced to combine the minor misspelling differences at the character level and both position and frequency parameters at the token level.

The experimental results demonstrate that taking position weight $\beta$ and character threshold $\theta$ into account to weight distance metrics improves the performance of name matching. The proper value for $\theta$ is critical since it determines the granularity level behavior of the algorithm. For example, if $\theta$ is set to one, the algorithm ignores all individual token spelling errors and tends to be a token level algorithm. On the other hand, setting $\theta$ to zero, the algorithm considers all token pairs to be matched with a similarity score; hence, the operation of the algorithm approaches character level algorithms. Sets that are characterized by more spelling errors need higher thresholds, while those that are characterized with token omissions need lower thresholds. Nevertheless, our presented hybrid name matching algorithm is able to work efficiently under different types of errors. Due to flexibility in changing the algorithm parameters, it can be optimized according to dataset characteristics to obtain better results.

Comparing the performance of different name matching algorithms is still a problematic issue. It was agreed that even metrics that demonstrate high performance for some data sets can perform poorly on others (Bilenko & Mooney, 2003). In a real environment, one cannot expect the matched set characterization. Therefore, in this work, we follow another comparison methodology by setting upper and lower limits of success of tested algorithms based on a variety of error schemes they may face in real sets. Clearly, highly spanned success algorithms are better and will have robust performance for different datasets. Also, since we intend to automate the whole matching process, we defined a 'success' of a tested algorithm as its ability to produce the correct matched name as a top scoring name. Using this methodology with a large Arabic dataset, the average performance of the proposed algorithm was compared with other classical algorithms. It was found that it raises the minimum success level from 69% – of best tested Soft TFIDF algorithm – to about 80%, (for second and third token omission), while achieving an upper accuracy limit of 99.67%. The best accuracy is lower than the basic Levenshtein algorithm, which achieves 100% upper accuracy levels for single and two character omission. Nevertheless, basic Levenshtein achieves only 16% as a lower accuracy level.

The concepts used in this algorithm are general and can be applied to name matching in many other languages in which the naming system is characterized by writing names in a restricted correct order, omission of one or more middle names, omission of non-significant components of names, and rare use of abbreviations for tokens. The only modification required to apply the proposed algorithm to other languages is the replacement of the Arabic name frequency table with the specific language one.

As a future work, we are working to upgrade the algorithm to deal with cross-language name matching. Name matching across languages requires approximate mapping of a name from one language into the other before applying a name matching algorithm. Nevertheless, the problem is not a one-to-one character (or phoneme) replacement of name components in both languages, since different writing styles should be considered. For example, the current algorithm has to be modified to deal with name abbreviations and a change of token order in other languages.

## References

Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., & Fienberg, S. (2003). Adaptive name matching in information integration. *Intelligent Systems, IEEE*, *18*(5), 16-23.

Bilenko, M., & Mooney, R. (2003). Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (2003)*, 39-48.

Borgman, C. L., & Siegfried, S. L. (1999). Getty's Synoname™ and its cousins: A survey of applications of personal name-matching algorithms. *Journal of the American Society for Information Science*, *43*(7), 459-476

Camacho, D., Huerta, R., & Elkan, C. (2008). An Evolutionary Hybrid Distance for Duplicate String Matching. http://arantxa.ii.uam.es/~dcamacho/ StringDistance/ hybrid-distance.pdf .

Cohen, W., Ravikumar, P., & Fienberg, S. (2003). A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03)*, 73-78.

Elmagarmid, A., Ipeirotis, P., & Verykios, V. (2007). Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *19*(1), 1-16.

Gadd, T. N. (1990). PHONIX: The algorithm. *Program: electronic library and information systems*, *24*(4), 363-366.

Hall, P., & Dowling, G. (1980). Approximate string matching. *ACM Computing Surveys (CSUR)*, *12*(4), 381-402.

Holmes, D., & McCabe, M. (2002). Improving precision and recall for soundex retrieval. In *Information Technology: Proceedings. International Conference on Coding and Computing 2002*, 22-26.

Jaro, M. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, *84*(406), 414-420.

Jurafsky, D., Martin, J., & Kehler, A. (2002). *Speech and language processing: an introduction to natural language processing, computational linguistics and speech recognition* (Vol. 2). Prentice Hall.

Keskustalo, H., Pirkola, A., Visala, K., Leppänen, E., & Järvelin, K. (2003). Non-adjacent digrams improve matching of cross-lingual spelling variants. In *String Processing and Information Retrieval, Lecture Notes in Computer Science* Volume 2857, 252-265. Springer Berlin/Heidelberg.

Monge, A., & Elkan, C. (1996). The field matching problem: Algorithms and applications. In *Proceedings of the second international Conference on Knowledge Discovery and Data Mining*, 267-270.

Moreau, E., Yvon, F., & Cappé, O. (2008). Robust similarity measures for named entities matching. In *Proceedings of the 22nd International Conference on Computational Linguistics*, *1*, 593-600.

Patman, F., & Thompson, P. (2003). Names: A new frontier in text mining. *Intelligence and Security Informatics, Lecture Notes in Computer Science* Volume 2665, 960-960.

Smith, T., & Waterman, M. (1981). ªIdentification of Common Molecular Subsequences. º *J. Molecular Biology*, *147*, 195-197.

Snae, C. (2007). A comparison and analysis of name matching algorithms. *International Journal of Applied Science, Engineering and Technology*, *4*(1), 252-257.

Waterman, M., Smith, T., & Beyer, W. (1976). Some biological sequence metrics. *Advances in Mathematics*, *20*(3), 367-387.

Winkler, W. (2002). Methods for record linkage and Bayesian networks. Technical report, *Statistical Research Division*, (2002). US Census Bureau, Washington, DC.

Winkler, W. (2006). Overview of record linkage and current research directions. In Bureau of the Census. (2006).

Xiao, C., Wang, W., Lin, X., Yu, J., & Wang, G. (2011). Efficient similarity joins for near-duplicate detection. *ACM Transactions on Database Systems (TODS)*, *36*(3), 15.

Computational Linguistics and Chinese Language Processing
Vol. 19, No. 1, March 2014, pp. 55-68

# On the Use of Speech Recognition Techniques to Identify Bird Species

Wei-Ho Tsai* and Yu-Zhi Xue*

## Abstract

Wild bird watching has become a popular leisure activity in recent years. Very often, people can see birds or hear their sounds, but have no idea what kind of bird species they are seeing. To help people learn to identify bird species from their sounds, we apply speech recognition techniques to build an automatic bird sound identification system. In this system, two acoustic cues are used for analysis, timbre and pitch. In the timbre-based analysis, Mel-Frequency Cepstral Coefficients (MFCCs) are used to characterize the bird sound. Then, we use Gaussian Mixture Models to represent the MFCCs as a set of parameters. In the pitch-based analysis, we convert bird sounds from their waveform representations into a sequence of MIDI notes. Then, Bigram models are used to capture the dynamic change information of the notes. We chose the top ten common bird species in the Taipei urban area to examine our system. Experiments conducted using audio data collected from commercial CDs and websites show that the timbre-based, pitch-based, and the combination thereof systems achieve 71.1%, 72.1%, and 75.0% accuracy of bird sound identification, respectively.

**Keywords:** Bird Species Identification, Bigram Model, Gaussian Mixture Model, Pitch, Timbre

## 1. Introduction

There are more than nine thousand and seven hundred bird species in the world. Although a number of birds are commonly seen, most people cannot recognize any of them. In this study, we attempt to develop automated techniques for identifying bird species from their sounds. Hereafter, this problem is referred to as bird sound identification. It is hoped that the

* Department of Electronic Engineering & Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, No.1, Sec. 3, Chunghsiao E. Rd. Taipei City, 10608, Taiwan, Tel.: +886-2-27712171; Fax: +886-2-27317120
E-mail: whtsai@ntut.edu.tw
The author for correspondence is Wei-Ho Tsai.

techniques can help people learn about such animals by simply recording the bird sounds they hear and sending the recording to our system.

Up to now, there has been very limited published research devoted to bird sound identification. In (Anderson *et al.,* 1996), Anderson *et al.* used dynamic time warping to measure the differences in spectrogram between an unknown bird sound recording and the template bird sound recordings. In (Kogan & Margoliash, 1998), Kogan *et al.* compared the performance of bird sound identification obtained with dynamic time warping and hidden Markov model, in which six acoustic features were used: linear predictive coding coefficients (LPCs), LPC-derived cepstral coefficients, LPC reflection, Mel-Frequency Cepstral Coefficients (MFCCs), log mel-filter bank channel, and linear mel-filter bank channel. In (McIlraith & Card, 1997), McIlraith *et al.* used a backpropagation neural network and multivariate statistics to perform bird sound identification. The acoustic features tested in (McIlraith & Card, 1997) are the number of syllables, average syllable duration, standard deviation of syllable durations, average pause duration, and standard deviation of pause durations. In (Somervuo *et al.,* 2006), Somervuo *et al.* compared three acoustic features on bird sound identification: sinusoidal modeling features, MFCCs, and descriptive features. Nevertheless, it is worth noting that all of the aforementioned studies tackle bird sound identification from the perspective of timbre-based analysis only. They all ignore bird sounds' pitch information, which is an important factor in why a bird sound is often called a bird song.

In this work, we propose a bird sound identification system based on timbre and pitch analyses. In addition to applying the most prevalent speaker-identification method to our system, we devise a method for exploiting the pitch information in bird sounds. Our experiments show that bird sound identification based on pitch information performs slightly better than that based on timbre information. It is further observed that combined use of timbre and pitch information achieves superior performance over the use of the individual information.

The remainder of this paper is organized as follows. Section 2 introduces the configuration of the proposed bird sound system, in which the two major components, timbre-based analysis and pitch-based analysis, are described in Sections 3 and 4, respectively. Section 5 discusses the experiments for examining our system. In Section 6, we present the conclusions and direction of our future works.

## 2. System Overview

Figure 1 shows the proposed bird sound identification system. In essence, the system can be divided into two components, namely timbre-based analysis and pitch-based analysis. Both components operate in two phases: training and testing. The purpose of the training phase is to extract the timbre and pitch features in each bird species' sound and to represent the features as two sets of parametric models. In the testing phase, the system takes as input an unknown sound recording and produces as output two likelihood scores from the timbre-based and pitch-based analyses, respectively. The scores then are combined to serve as the basis of the decision. According to the maximum likelihood decision rule, the system decides an unknown sound recording in favor of bird species $B^*$ when the condition in Eq. (1) is satisfied:

$$B^* = \arg\max_{1 \le i \le N}(\alpha \cdot v_i + \beta \cdot r_i)\,, \tag{1}$$

where $N$ is the number of bird species; $v_i$ and $r_i$ are the likelihood scores output from the timbre-based and pitch-based analyses with respect to the $i$-th bird species' models, respectively; and $\alpha$ and $\beta$ are tunable weights.

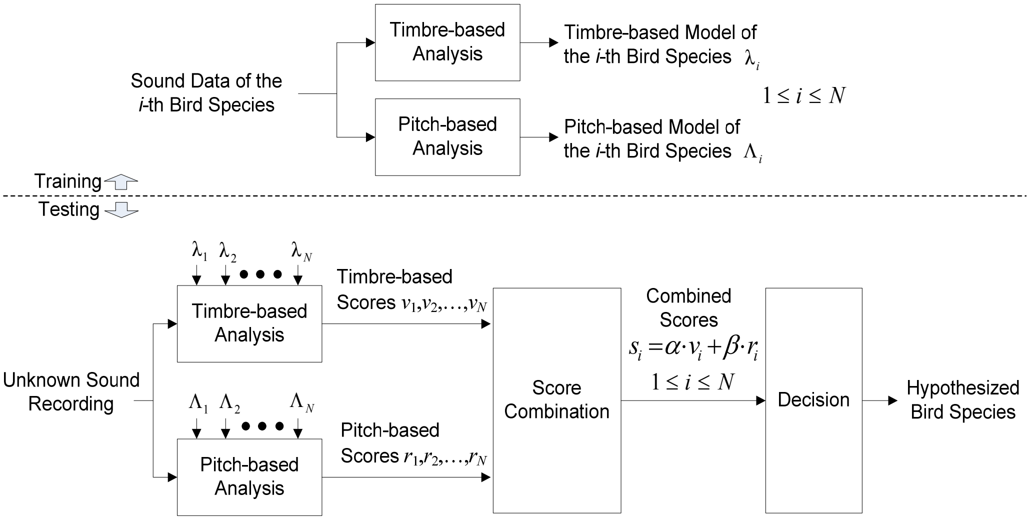

Sound Data of the
i-th Bird Species
Timbre-based Analysis
Pitch-based Analysis
Timbre-based Model of the i-th Bird Species $\lambda_i$
Pitch-based Model of the i-th Bird Species $\Lambda_i$
$1 \le i \le N$
Training
Testing
$\lambda_1$ $\lambda_2$ $\lambda_N$
$\Lambda_1$ $\Lambda_2$ $\Lambda_N$
Unknown Sound Recording
Timbre-based Scores $v_1, v_2, \ldots, v_N$
Pitch-based Scores $r_1, r_2, \ldots, r_N$
Score Combination
Combined Scores
$s_i = \alpha \cdot v_i + \beta \cdot r_i$
$1 \le i \le N$
Decision
Hypothesized Bird Species

***Figure 1. The proposed bird sound identification system.***

## 3. Timbre-based Analysis

Figure 2 shows the procedure of the timbre-based analysis. It consists of feature extraction and Gaussian mixture modeling in the training phase, along with feature extraction and likelihood computation in the testing phase.

### 3.1 Feature Extraction

Among the timbre-based features investigated in (Kogan & Margoliash, 1998), the Mel-scale Frequency Cepstral Coefficients (MFCCs) feature (Davis & Mermelstein, 1980) has been found to be superior to the others in bird sound identification. To compute MFCCs, a waveform signal first is divided into frames using a $P$-length sliding Hamming window with $0.5P$-length overlapping between frames. Every frame then undergoes Hamming windowing and fast Fourier transform (FFT) with size *J*. Next, each frame is passed through a set of triangular filter banks, equally spaced on a Mel scale. Let $|A_{t,j}|$ denote the signal's magnitude with respect to FFT index *j* in frame *t*, where $1 \le j \le J$. Then,

$$X_{t,i} = \frac{1}{B}\sum_{b=1}^{B}\left\{\log\left(\sum_{j=l_b}^{u_b}\left|A_{t,j}\right|^2 T_b(j)\right)\cdot\cos\left(\frac{\pi i}{B}(b-0.5)\right)\right\}, 1 \le i \le B\,, \tag{2}$$

where *B* is the total number of filter banks, $l_b$ is the lowest frequency index in the *b*-th bank, $u_b$ is the highest frequency index in the *b*-th bank, and $T_b(j)$ is the response of the *b*-th bank. Briefly, MFCCs represent the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. It is found that the nonlinear mel scale of frequency approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the regular cepstrum.

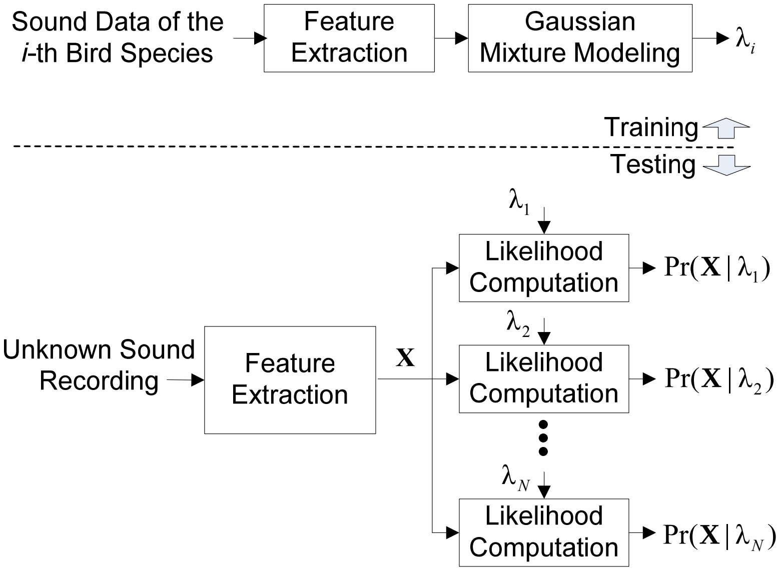


***Figure 2. The procedure of the timbre-based analysis.***

## 3.2 Gaussian Mixture Modeling

To capture the collective sound characteristics of each bird species, all of the MFCCs of each bird species are pooled together to form a Gaussian mixture model (GMM) (Reynolds & Rose, 1995). It is assumed that each bird species has its own timbre pattern that reflects in the distribution of MFCCs over a span of time. A GMM approximates the static timbre patterns by a mixture of Gaussian densities. Note that the reason we capture the static timbre patterns rather than dynamic timbre patterns using hidden Markov models (HMMs) (Rabiner, 1989) is to prevent the resulting models from dependence on bird individuals or bird messages.

The parameters of a GMM consist of means, covariances, and mixture weights, which are commonly estimated using the Expectation-Maximization (EM) algorithm (Dempster *et al.,* 1977). Nevertheless, recognizing that the numbers of each bird species' sound samples for training may not be sufficient always, we use the GMM-MAP approach (Reynolds & Quatieri, 2000) to generate each bird species' GMM. Specifically, all of the MFCCs of all of the bird species first are pooled together to form a universal GMM using the EM algorithm. Then, the parameters of the universal GMM are modified with respect to each bird species using the MFCCs of the individual bird species based on maximum *a posteriori* (MAP) estimation. If there are *N* bird species to be identified, we generate *N* GMMs, $\lambda_1, \lambda_2, \cdots, \lambda_N$ .

## 3.3 Likelihood Computation

Given an unknown bird sound recording, the system computes its MFCCs $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_T\}$ before computing the likelihood probability $\Pr(\mathbf{X}|\lambda_j)$ for each model $\lambda_j$:

$$\Pr(\mathbf{X} \mid \lambda_j) = \prod_{t=1}^{T} \sum_{k=1}^{K} w_{j,k} \cdot \frac{1}{\pi^N \left|\mathbf{C}_{j,k}\right|} \exp\left\{ -\left(\mathbf{X}_t - \boldsymbol{\mu}_{j,k}\right)' \mathbf{C}_{j,k}^{-1} \left(\mathbf{X}_t - \boldsymbol{\mu}_{j,k}\right) \right\}, \tag{3}$$

where $K$ is the number of mixture Gaussian components; $w_{j,k}$, $\boldsymbol{\mu}_{j,k}$, and $\mathbf{C}_{j,k}$ are the $k$-th mixture weight, mean, and covariance of model $\lambda_j$, respectively; and prime (′) denotes the vector transpose.

## 4. Pitch-based Analysis

As bird sound is often regarded as a type of music, it is reasonable to assume that each bird species has its own pitch pattern that can be exploited to distinguish from other species. Pitch is the reciprocal of fundamental frequency; hence, a bird sound recording can be viewed as a sequence of fundamental frequencies. We then can model the variations of the fundamental frequencies to characterize each bird species' sounds. Nevertheless, considering that the estimation of fundamental frequency is prone to numerical errors, we use MIDI note numbers instead of fundamental frequencies to explore the pitch information in bird sounds. The MIDI note numbers can be treated as the non-linear quantization of fundamental frequencies and can absorb the numerical errors during the estimation of fundamental frequencies. Figure 3 shows the procedure of pitch-based analysis. It consists of MIDI note extraction for converting sound recordings from waveform representations into MIDI note sequences and bigram modeling for characterizing the underlying pitch information in the note sequences.

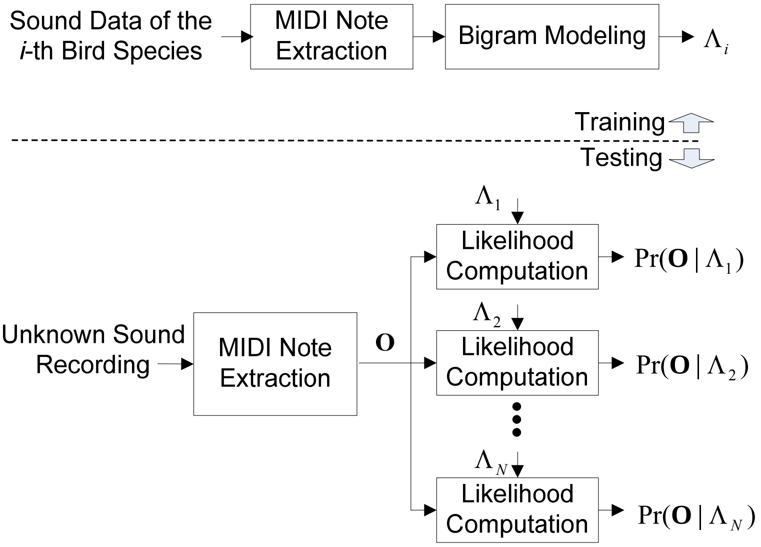


***Figure 3. The procedure of pitch-based analysis.***

## 4.1 MIDI Note Extraction

Let $e_m$, $1\leq m \leq M$, be the inventory of possible notes produced by a bird. Our aim is to determine which among the $M$ possible notes is most likely produced at each instant in a bird sound recording. We apply the strategy in (Yu *et al.,* 2008) to solve this problem. First, the bird sound is divided into frames using a $P$-length sliding Hamming window, with $0.5P$-length overlapping between frames. Every frame then undergoes a Fast Fourier Transform (FFT) with size $J$. Let $x_{t,j}$ denote the signal's energy with respect to FFT index $j$ in frame $t$, where $1 \leq j\leq J$, and $x_{t,j}$ has been normalized to the range between 0 and 1. Then, the signal's energy on the $m$-th note in frame $t$ can be estimated by:

$$\hat{x}_{t,m} = \max_{\forall j, U(j)=e_m} x_{t,j} \quad , \tag{4}$$

and

$$U(j) = \left\lfloor 12 \cdot \log_2\left(\tfrac{F(j)}{440}\right) + 69.5 \right\rfloor, \tag{5}$$

where $\lfloor\ \rfloor$ is a floor operator, $F(j)$ is the corresponding frequency of FFT index $j$, and $U(\cdot)$ represents a conversion between the FFT indices and the MIDI note numbers.

Ideally, if note $n_m$ is sung in frame $t$, the resulting energy, $\hat{x}_{t,m}$, should be the maximum among $\hat{x}_{t,1}, \hat{x}_{t,2}, \ldots, \hat{x}_{t,M}$. Nevertheless, it is sometimes the case that the energy of a true note is smaller than that of its harmonic note. To avoid the interference of harmonics in the estimation of true notes, we use the strategy of Sub-Harmonic Summation (SHS) (Piszczalski & Galler, 1979), which computes a value for the "strength" of each possible note by summing the signal's energy on a note and its harmonic note numbers. Specifically, the strength of note $n_m$ in frame $t$ is computed using

$$y_{t,m} = \sum_{c=0}^{C} h^c \hat{x}_{t,m+12c} , \tag{6}$$

where $C$ is the number of harmonics considered, and $h$ is a positive value less than 1 that discounts the contribution of higher harmonics. The result of this summation is that the true note usually receives the largest amount of energy from its harmonic notes. Thus, the true note in frame $t$ can be determined by choosing the note number associated with the largest value of the strength. Nevertheless, recognizing that a note usually lasts several frames, the decision could be made by including the information from neighboring frames. Specifically, we determine the sung note in frame $t$ by choosing the note number associated with the largest value of the strength accumulated for adjacent frames, *i.e.*,

$$o_t = \arg\max_{1 \le m \le M} \sum_{b=-W}^{W} y_{t+b,m} , \tag{7}$$

Further, the resulting note sequence is refined by taking into account the continuity between frames. This is done with median filtering, which replaces each note with the local median of notes of its neighboring ±$W$ frames to remove jitters between adjacent frames. In the implementation, the range of $e_m$ is set to be $60 \le e_m \le 120$, corresponding to fundamental frequency from to 261.6 to 8591 Hz.

## 4.2 Bigram Modeling

After converting bird sounds into sequences of MIDI notes, we use a bigram model (Huang *et al.,* 2001) to capture the dynamic information in the note sequences. The bigram model consists of a set of bigram probabilities and unigram probabilities. The bigram probabilities $\Pr(e_j|e_i)$, $1 \le i, j \le M$, account for the frequency of a certain note $e_i$ followed by another note $e_j$, while the unigram probabilities $\Pr(e_i)$ account for the frequency of occurring a certain note $e_i$. It is assumed that each bird species has its own pitch pattern that reflects in the frequency of occurrence of one or a pair of notes. For $N$ bird species to be identified, we generate $N$ bigram models $\Lambda_1, \Lambda_2, \ldots, \Lambda_N$.

## 4.3 Likelihood Computation

In the testing phase, an unknown bird sound recording first is converted into a sequence of notes $\mathrm{O} = \mathrm{o}_1, \mathrm{o}_2, \cdots, \mathrm{o}_T$, then tested against each bigram model $\Lambda_i, 1 \le i \le N$. The results of testing are likelihood probabilities:

$$\Pr(\mathbf{O} \mid \Lambda) = \Pr(o_1) \cdot \prod_{t=2}^{T} \Pr(o_t \mid o_{t-1}) . \tag{8}$$

# 5. Experiments

## 5.1 Bird Sound Data

The bird sound data used in this study stem from the commercial CDs and websites listed in Table 1. To facilitate the experiments, all of the sound data were converted into PCM WAV with 22.05-kHz sampling rate and 16-bit quantization resolution. We chose ten bird species commonly seen in the Taipei urban area, including *Dicrurus aeneus*, *Dendrocopos canicapillus*, *Pomatorhinus ruficollis*, *Stachyris ruficeps*, *Megalaima oorti*, *Heterophasia auricularis*, *Hypsipetes madagascariensis*, *Myiophonus insularis*, *Otus spilocephalus*, and *Dendrocitta formosae*. The data were divided into two subsets, training and testing. The amount of sound data with respect to each bird species is listed in Table 2.

***Table 1. Source of our bird sound data***

| **CDs or Websites** | **Audio Types** |
|---|---|
| "Songbirds" CDs published by WIND RECORDS CO., LTD. | 44.1kHz Sampling Rate<br>16-bit Quantization Resolution CDs |
| "Birdwatcher's guide to the Taipei region" CDs published by Department of Information and Tourism,Taipei City Government | 44.1kHz Sampling Rate<br>16-bit Quantization Resolution CDs |
| http://archive.zo.ntu.edu.tw/ | 44.1kHz Sampling Rate<br>16-bit Quantization Resolution WAV Files |
| http://macaulaylibrary.org/index.do | Streaming Audio |

***Table 2. The amount of sound data with respect to each bird species***

| **Bird Species** | **Training Data:**<br>Total Duration (sec) | **Testing Data:**<br>No. of Sound Samples |
|---|---|---|
| *Dicrurus aeneus* | 130 | 77 |
| *Dendrocopos canicapillus* | 35 | 102 |
| *Pomatorhinus ruficollis* | 125 | 155 |
| *Stachyris ruficeps* | 66 | 81 |
| *Megalaima oorti* | 93 | 219 |
| *Heterophasia auricularis* | 91 | 86 |
| *Hypsipetes madagascariensis* | 42 | 157 |
| *Myiophonus insularis* | 27 | 25 |
| *Otus spilocephalus* | 27 | 111 |
| *Dendrocitta formosae* | 39 | 73 |

## 5.2 Experiment Results

Our experiments were conducted to examine the timbre-based component and pitch-based component separately before evaluating if the performance of bird sound identification could be further improved by combining the two components. The performance was characterized with the accuracy:

$$\text{Accuracy (in\%)}=\frac{\text{Tonal number of correctly - identified recordings}}{\text{Tonal number of testing recordings}}\times 100\%$$

### 5.2.1 Accuracies Obtained with the Timbre-based Analysis

In the timbre-based analysis, the MFCC feature vectors, each consisting of 20 coefficients, were extracted from the bird sound data, using a 30-ms Hamming-windowed frame with 15-ms frame shifts. The FFT size was set to be 2048. Table 3 shows the identification accuracies obtained with various numbers of mixture Gaussian densities used in GMM. The best accuracy in Table 3 is 71.1%, achieved with 64 mixtures. Table 4 shows the confusion matrix of the identification for the case of 64 mixtures. We can see from Table 4 that the timbre-based analysis performs best in identifying *Pomatorhinus ruficollis*, whereas it performs worst in identifying *Dendrocitta formosae*.

***Table 3. Identification accuracies (in %) obtained with various numbers of mixture Gaussian densities used in GMM.***

| # mixtures | **4** | **8** | **16** | **32** | **64** | **128** |
|---|---|---|---|---|---|---|
| *Dicrurus aeneus* | 55.8 | 58.4 | 58.4 | 59.7 | 64.9 | 62.3 |
| *Dendrocopos canicapillus* | 59.8 | 59.8 | 60.8 | 62.7 | 62.7 | 62.7 |
| *Pomatorhinus ruficollis* | 80 | 81.9 | 83.2 | 83.2 | 82.6 | 81.9 |
| *Stachyris ruficeps* | 69.1 | 69.1 | 70.4 | 71.6 | 70.4 | 69.1 |
| *Megalaima oorti* | 72.1 | 73.1 | 74 | 74.4 | 74.9 | 74.9 |
| *Heterophasia auricularis* | 74.4 | 75.6 | 78 | 77.9 | 76.7 | 76.7 |
| *Hypsipetes madagascariensis* | 62.4 | 63.7 | 65 | 66.2 | 68.8 | 67.5 |
| *Myiophonus insularis* | 68 | 72 | 76 | 76 | 76 | 76 |
| *Otus spilocephalus* | 64 | 64 | 64 | 64 | 67.6 | 65.8 |
| *Dendrocitta formosae* | 50.7 | 52.1 | 56.2 | 57.5 | 56.2 | 54.8 |
| **Average Accuracy** | **67.1** | **68.2** | **69.5** | **70.3** | **71.1** | **70.3** |

***Table 4. Confusion matrix of the identification for the case of 64 mixtures.***

| Identified<br>True | *Dicrurus aeneus* | *Dendrocopos canicapillus* | *Pomatorhinus ruficollis* | *Stachyris ruficeps* | *Megalaima oorti* | *Heterophasia auricularis* | *Hypsipetes madagascariensis* | *Myiophonus insularis* | *Otus spilocephalus* | *Dendrocitta formosae* |
|---|---|---|---|---|---|---|---|---|---|---|
| *Dicrurus aeneus* | 64.9 | 10.4 | 0 | 9.1 | 0 | 6.5 | 6.5 | 0 | 2.6 | 0 |
| *Dendrocopos canicapillus* | 9.8 | 62.7 | 14.7 | 2.9 | 3.9 | 2.9 | 2.9 | 0 | 0 | 0 |
| *Pomatorhinus ruficollis* | 0 | 6.5 | 82.6 | 0 | 0 | 0 | 0 | 6.5 | 0 | 4.5 |
| *Stachyris ruficeps* | 7.4 | 3.7 | 6.2 | 70.4 | 6.2 | 3.7 | 0 | 2.5 | 0 | 0 |
| *Megalaima oorti* | 0 | 0.9 | 0 | 0 | 74.9 | 0 | 13.7 | 0 | 6.8 | 3.7 |
| *Heterophasia auricularis* | 3.9 | 0 | 8.1 | 0 | 0 | 76.7 | 5.8 | 5.8 | 0 | 0 |
| *Hypsipetes madagascariensis* | 0 | 6.4 | 0 | 4.5 | 8.3 | 0 | 68.8 | 2.5 | 15.9 | 0 |
| *Myiophonus insularis* | 0 | 0 | 16 | 0 | 0 | 8 | 0 | 76 | 0 | 0 |
| *Otus spilocephalus* | 2.7 | 13.5 | 0 | 9 | 0 | 7.2 | 0 | 0 | 67.6 | 0 |
| *Dendrocitta formosae* | 2.7 | 0 | 11 | 0 | 0 | 0 | 21.9 | 0 | 8.2 | 56.2 |

### 5.2.2 Accuracies Obtained with the Pitch-based Analysis

We then tested the pitch-based analysis component. The length of frame and FFT size were the same as the settings in computing MFCCs. Table 5 shows the resulting confusion matrix of the identification. We obtained an average identification accuracy of 72.0%, which is slightly higher than that obtained with the timbre-based analysis. Comparing Tables 4 and 5, we can see that the misidentified cases for timbre-based analysis and pitch-based analysis are different. This indicates that combined use of the two components would achieve higher identification accuracy than the use of an individual component.

### 5.2.3 Combined Use of the Timbre-based and Pitch-based Analyses

Finally, we examined the proposed system based on the combination of timbre-based analysis and pitch-based analysis. Table 6 shows the identification accuracies obtained with different settings in the value of $\alpha$ and $\beta$. We can see from Table 6 that the combined use of timbre-based analysis and pitch-based analysis does perform better than both timbre-based analysis and pitch-based analysis used solely. It also can be seen that the resulting accuracies are not sensitive to the values of $\alpha$ and $\beta$, as long as they are set to a certain range. Table 7 shows the confusion matrix of the identification for the case of $\alpha$ = 0.4 and $\beta$ = 0.6, which achieves an average accuracy of 75.0%. We can see from Table 7 that the overall system improves the accuracies of identifying almost every bird species, compared to Tables 4 and 5. This result confirms the validity of the proposed system.

***Table 5. Confusion matrix of the identification using pitch-based analysis.***

| Identified / True | *Dicrurus aeneus* | *Dendrocopos canicapillus* | *Pomatorhinus ruficollis* | *Stachyris ruficeps* | *Megalaima oorti* | *Heterophasia auricularis* | *Hypsipetes madagascariensis* | *Myiophonus insularis* | *Otus spilocephalus* | *Dendrocitta formosae* |
|---|---|---|---|---|---|---|---|---|---|---|
| *Dicrurus aeneus* | 61 | 19.9 | 0 | 10.4 | 0 | 5.2 | 3.9 | 0 | 0 | 0 |
| *Dendrocopos canicapillus* | 2.9 | 71.6 | 12.7 | 0 | 2 | 0 | 3.9 | 0 | 2.9 | 3.9 |
| *Pomatorhinus ruficollis* | 0 | 7.7 | 82.9 | 0 | 0 | 0 | 0 | 7.7 | 0 | 1.9 |
| *Stachyris ruficeps* | 1.2 | 0 | 7.4 | 75.3 | 2.5 | 1.2 | 0 | 0 | 0 | 0 |
| *Megalaima oorti* | 0 | 1.4 | 0 | 1.8 | 82.2 | 0 | 11.4 | 0 | 2.7 | 0.5 |
| *Heterophasia auricularis* | 0 | 0 | 11.6 | 0 | 0 | 76.7 | 5.8 | 5.8 | 0 | 0 |
| *Hypsipetes madagascariensis* | 0 | 6.4 | 0 | 2.5 | 9.6 | 0 | 63.1 | 2.5 | 15.9 | 0 |
| *Myiophonus insularis* | 0 | 0 | 4 | 0 | 12 | 28 | 0 | 56 | 0 | 0 |
| *Otus spilocephalus* | 7.2 | 21.6 | 5.4 | 0 | 0 | 0.1 | 0 | 0 | 64.9 | 0 |
| *Dendrocitta formosae* | 5.4 | 0 | 8.2 | 0 | 0 | 0 | 24.7 | 0 | 2.7 | 58.9 |

***Table 6. Accuracies (in %) obtained with different settings in the value of α and β.***

| α β | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|
| 0.9 | 73.3 | - | - | - | - | - | - | - | - |
| 0.8 | - | 74.0 | - | - | - | - | - | - | - |
| 0.7 | - | - | 74.1 | - | - | - | - | - | - |
| 0.6 | - | - | - | 74.6 | - | - | - | - | - |
| 0.5 | - | - | - | - | 74.9 | - | - | - | - |
| 0.4 | - | - | - | - | - | 75 | - | - | - |
| 0.3 | - | - | - | - | - | - | 74.8 | - | - |
| 0.2 | - | - | - | - | - | - | - | 74.7 | - |
| 0.1 | - | - | - | - | - | - | - | - | 73.1 |

***Table 7. Confusion matrix of the identification for the case of α = 0.4 and β = 0.6***

| Identified<br>True | *Dicrurus aeneus* | *Dendrocopos canicapillus* | *Pomatorhinus ruficollis* | *Stachyris ruficeps* | *Megalaima oorti* | *Heterophasia auricularis* | *Hypsipetes madagascariensis* | *Myiophonus insularis* | *Otus spilocephalus* | *Dendrocitta formosae* |
|---|---|---|---|---|---|---|---|---|---|---|
| *Dicrurus aeneus* | 67.5 | 13 | 0 | 9.1 | 0 | 5.2 | 2.6 | 0 | 2.6 | 0 |
| *Dendrocopos canicapillus* | 2.9 | 75.5 | 9.8 | 0 | 0.1 | 0 | 3.9 | 0 | 2.9 | 3.9 |
| *Pomatorhinus ruficollis* | 0 | 5.8 | 85.2 | 0 | 0 | 0 | 0 | 5.8 | 0 | 3.2 |
| *Stachyris ruficeps* | 1.2 | 0 | 6.2 | 75.3 | 2.5 | 1.2 | 0 | 1.2 | 0 | 0 |
| *Megalaima oorti* | 0 | 1.4 | 0 | 1.8 | 83.1 | 0 | 9.1 | 0 | 3.2 | 1.4 |
| *Heterophasia auricularis* | 0 | 0 | 10.5 | 0 | 0 | 80.2 | 4.7 | 3.5 | 0 | 0 |
| *Hypsipetes madagascariensis* | 0 | 6.4 | 0 | 1.3 | 9.6 | 0 | 65.6 | 1.3 | 15.9 | 0 |
| *Myiophonus insularis* | 0 | 0 | 4 | 0 | 4 | 12 | 0 | 80 | 0 | 0 |
| *Otus spilocephalus* | 7.2 | 21.6 | 5.4 | 0 | 0 | 1.8 | 0 | 0 | 64 | 0 |
| *Dendrocitta formosae* | 4.1 | 0 | 5.5 | 0 | 0 | 0 | 21.9 | 0 | 2.7 | 65.8 |

## 6. Conclusion

This work has developed an automatic bird sound identification system, with the motivation of helping people learn to identify bird species from their sounds. The system is built on speech recognition techniques, along with specific tailoring to handle the bird sound characteristics. Two acoustic cues were investigated for analysis, timbre and pitch. In the timbre-based analysis, we used MFCCs to characterize the bird sound. Then, GMMs were used to represent the MFCCs as a set of parameters. In the pitch-based analysis, we converted bird sounds from their waveform representations into a sequence of MIDI notes. Then, Bigram models were used to capture the dynamic change information of the notes. Our experiments, conducted using audio data of the ten most common bird species in the Taipei urban area, show that the timbre-based, pitch-based, and the combined system achieves 71.1%, 72.1%, and 75.0% accuracy of bird sound identification, respectively.

Despite the potential, the performance of the proposed bird sound identification system still leaves considerable room for improvement. In the future, we will try to include more characteristics of bird sounds, such as the concept of bird calls and bird songs, into our system design. In addition, we have to scale up our sound database to hundreds or thousands of bird species to validate the proposed identification system.

## Acknowledgement

This work was supported in part by the National Science Council, Taiwan, under Grant No. NSC 99-2628-E-027-005.

## References

Anderson, S. E., Dave, A. S., & Margoliash, D. (1996) .Template-based automatic recognition of birdsong syllables from continuous recordings. *J. Acoust. Soc. Amer.*, *100*(2), 1209-1219.

Davis, S. B., & Mermelstein, P. (1980). Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Trans. on Acoustic, Speech and Signal Processing.*, *28*(4), 357-366.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc.*, *39*, 1-38.

Huang, X., Acero, A., & Hon, H. W. (2001). *Spoken Language Processing*, Prentice Hall.

Kogan, J., & Margoliash, D. (1998). Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden Markov models: A comparative study. *J. Acoust. Soc. Amer.*, *103*(4), 2187-2196.

McIlraith, A. L., & Card, H. C. (1997). Birdsong recognition using backpropagation and multivariate statistics. *IEEE Trans. Signal Process.*, *45*(11), 2740-2748.

Piszczalski, M., & Galler, B. A. (1979). Predicting musical pitch from component frequency ratios. *Journal of the Acoustical Society of America*, *66*(3), 710-720.

Rabiner, L. R. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257-286.

Reynolds, D., & Rose, R. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Trans. Speech Audio Process.*, *3*(1), 72-83.

Reynolds, D., & Quatieri, T. (2000). Speaker Verification Using Adapted Gaussian ixture Models. *Digital Signal Processing, 10*, 19-41.

Somervuo, P., Härmä, A., & Fagerlund, S. (2006). Parametric representations of bird sounds for automatic species recognition. *IEEE Trans. Audio, Speech, Language Process.*, *14*(6), 2252-2263.

Yu, H. M., Tsai, W. H., & Wang, H. M. (2008). A query-by-Singing system for retrieving karaoke music. *IEEE Trans. Multimedia*, *10*(8), 1626-1637.

# The Association for Computational Linguistics and Chinese Language Processing

(new members are welcomed)

## Aims：

1. To conduct research in computational linguistics.
2. To promote the utilization and development of computational linguistics.
3. To encourage research in and development of the field of Chinese computational linguistics both domestically and internationally.
4. To maintain contact with international groups who have similar goals and to cultivate academic exchange.

## Activities：

1. Holding the Republic of China Computational Linguistics Conference (ROCLING) annually.
2. Facilitating and promoting academic research, seminars, training, discussions, comparative evaluations and other activities related to computational linguistics.
3. Collecting information and materials on recent developments in the field of computational linguistics, domestically and internationally.
4. Publishing pertinent journals, proceedings and newsletters.
5. Setting of the Chinese-language technical terminology and symbols related to computational linguistics.
6. Maintaining contact with international computational linguistics academic organizations.
7. Dealing with various other matters related to the development of computational linguistics.

## To Register：

Please send application to:

The Association for Computational Linguistics and Chinese Language Processing
Institute of Information Science, Academia Sinica
128, Sec. 2, Academy Rd., Nankang, Taipei 11529, Taiwan, R.O.C.

payment： Credit cards(please fill in the order form), cheque, or money orders.

## Annual Fees：

| | |
|---|---|
| regular/overseas member： | NT$ 1,000 (US$50.-) |
| group membership： | NT$20,000 (US$1,000.-) |

life member：ten times the annual fee for regular/ group/ overseas members

## Contact：

Address： The Association for Computational Linguistics and Chinese Language Processing
Institute of Information Science, Academia Sinica
128, Sec. 2, Academy Rd., Nankang, Taipei 11529, Taiwan, R.O.C.

Tel.：886-2-2788-3799 ext. 1502 Fax：886-2-2788-1638

E-mail: aclclp@hp.iis.sinica.edu.tw Web Site: http://www.aclclp.org.tw

Please address all correspondence to Miss Qi Huang, or Miss Abby Ho

# The Association for Computational Linguistics and Chinese Language Processing

## Membership Application Form

Member ID#： ________________

Name： __________________________________ Date of Birth： _______________

Country of Residence： ____________________ Province/State： ____________________

Passport No.： ____________________________ Sex: ________________________

Education(highest degree obtained)： ________________________________________

Work Experience： __________________________________________________

__________________________________________________

Present Occupation： __________________________________________________

Address： ______________________________________________________________

______________________________________________________________

Email Add： ______________________________________________________________

Tel. No： ______________________________ Fax No： ___________________________

Membership Category：□ Regular Member □ Life Member

Date： _____/____/____（Y-M-D）

Applicant's Signature：

Remarks： Please indicated clearly in which membership category you wish to register, according to the following scale of annual membership dues：

| | | |
|---|---|---|
| Regular Member | ： | US$ 50.- （NT$ 1,000） |
| Life Member | ： | US$500.-（NT$10,000） |

Please feel free to make copies of this application for others to use.

Committee Assessment：

# 中華民國計算語言學學會

宗旨：

（一） 從事計算語言學之研究
（二） 推行計算語言學之應用與發展
（三） 促進國內外中文計算語言學之研究與發展
（四） 聯繫國際有關組織並推動學術交流

活動項目：

（一）定期舉辦中華民國計算語言學學術會議（Rocling）

（二）舉行有關計算語言學之學術研究講習、訓練、討論、觀摩等活動項目

（三）收集國內外有關計算語言學知識之圖書及最新發展之資料

（四）發行有關之學術刊物，論文集及通訊

（五）研定有關計算語言學專用名稱術語及符號

（六）與國際計算語言學學術機構聯繫交流

（七）其他有關計算語言發展事項

報名方式：

1. 入會申請書：請至本會網頁下載入會申請表，填妥後郵寄或E-mail至本會

2. 繳交會費：劃撥：帳號：19166251，戶名：中華民國計算語言學學會
信用卡：請至本會網頁下載信用卡付款單

年費：

| | | |
|---|---|---|
| 終身會員： | 10,000.- | （US$ 500.-） |
| 個人會員： | 1,000.- | （US$ 50.-） |
| 學生會員： | 500.- | （限國內學生） |
| 團體會員： | 20,000.- | （US$ 1,000.-） |

連絡處：

地址：台北市115南港區研究院路二段128號 中研院資訊所(轉)
電話：(02) 2788-3799 ext.1502 傳真：(02) 2788-1638
E-mail：aclclp@hp.iis.sinica.edu.tw 網址: http://www.aclclp.org.tw
連絡人：黃琪 小姐、何婉如 小姐

# 中華民國計算語言學學會
# 個人會員入會申請書

| | | | | | |
|---|---|---|---|---|---|
| 會員類別 | □終身 □個人 □學生 | 會員編號 | | | （由本會填寫） |
| 姓名 | | 性別 | | 出生日期 | 年　月　日 |
| | | | | 身分證號碼 | |
| 現職 | | 學歷 | | | |
| 通訊地址 | □□□ | | | | |
| 戶籍地址 | □□□ | | | | |
| 電話 | | E-Mail | | | |
| | | | 申請人: | | （簽章） |
| | | | 中華民國 | 年 | 月　日 |

## 審查結果:

1. 年費：

   終身會員：　10,000.-
   個人會員：　1,000.-
   學生會員：　500.-（限國內學生）
   團體會員：　20,000.-

2. 連絡處：

   地址：台北市南港區研究院路二段128號 中研院資訊所(轉)
   電話：(02) 2788-3799　ext.1502 傳真：(02) 2788-1638
   E-mail：aclclp@hp.iis.sinica.edu.tw　網址: http://www.aclclp.org.tw
   連絡人：黃琪 小姐、何婉如 小姐

3. 本表可自行影印

# The Association for Computational Linguistics and Chinese Language Processing (ACLCLP)
# PAYMENT FORM

Name: ____________________________________(Please print) Date: _______________

**Please debit my credit card as follows:** US$ _____________

□ VISA CARD □ MASTER CARD □ JCB CARD Issue Bank:___________________

Card No.: ________ -_________-________ -_________ Exp. Date:____________(M/Y)

3-digit code: _________ (on the back card, inside the signature area, the last three digits)

CARD HOLDER SIGNATURE: ____________________________________

Phone No.: __________________________E-mail: _____________________________________

Address: ________________________________________________________________________

## PAYMENT FOR

US$ _____ □ Computational Linguistics & Chinese Languages Processing (IJCLCLP)

Quantity Wanted: ______

US$ _____ □ Journal of Information Science and Engineering (JISE)

Quantity Wanted: ______

US$ _____ □ Publications:_________________________________

US$ _____ □ Text Corpora: _______________________________

US$ _____ □ Speech Corpora:______________________________

US$ _____ □ Others: __________________________________

US$ _________ □ Membership Fees □ Life Membership □ New Membership □Renew

US$ _________ = Total

**Fax 886-2-2788-1638 or Mail this form to:**

ACLCLP
% IIS, Academia Sinica
Rm502, No.128, Sec.2, Academia Rd., Nankang, Taipei 115, Taiwan

**E-mail: aclclp@hp.iis.sinica.edu.tw**
**Website: http://www.aclclp.org.tw**

# 中華民國計算語言學學會
# 信用卡付款單

姓名: ____________________________(請以正楷書寫)　　日期:：____________________

卡別：❑ VISA CARD　　❑ MASTER CARD ❑ JCB CARD　　發卡銀行：______________

信用卡號：__________-__________-__________-__________　　有效日期：__________(m/y)

卡片後三碼：______________（卡片背面簽名欄上數字後三碼）

持卡人簽名：______________________________________(簽名方式請與信用卡背面相同)

通訊地址：______________________________________________________________________

聯絡電話：____________________________E-mail：_________________________________

備註：為順利取得信用卡授權，請提供與發卡銀行相同之聯絡資料。

**付款內容及金額：**

NT$__________ ❑ 中文計算語言學期刊(IJCLCLP) _______________

NT$__________ ❑ Journal of Information Science and Engineering (JISE)

NT$__________ ❑ 中研院詞庫小組技術報告_____________________

NT$__________ ❑ 文字語料庫 ____________________________________

NT$__________ ❑ 語音資料庫 ____________________________________

NT$__________ ❑ 光華雜誌語料庫1976~2010

NT$__________ ❑ 中文資訊檢索標竿測試集/文件集

NT$__________ ❑ 會員年費：❑續會　　　❑新會員　　　❑終身會員

NT$__________ ❑ 其他: _____________________________________

NT$_______________ ＝　合計

**填妥後請傳真至 02-27881638 或郵寄至:**
**11529台北市南港區研究院路2段128號中研院資訊所(轉)中華民國計算語言學學會 收**
**E-mail: aclclp@hp.iis.sinica.edu.tw**
**Website: http://www.aclclp.org.tw**

# Publications of the Association for Computational Linguistics and Chinese Language Processing

| | | | AIR | AIR | | |
|---|---|---|---|---|---|---|
| | | Surface | (US&EURP) | (ASIA) | VOLUME | AMOUNT |
| 1. | no.92-01, no. 92-04(合訂本) ICG 中的論旨角色與 A Conceptual Structure for Parsing Mandarin -- Its Frame and General Applications-- | US$ 9 | US$ 19 | US$15 | _____ | _____ |
| 2. | no.92-02 V-N 複合名詞討論篇<br>& 92-03 V-R 複合動詞討論篇 | 12 | 21 | 17 | _____ | _____ |
| 3. | no.93-01 新聞語料庫字頻統計表 | 8 | 13 | 11 | _____ | _____ |
| 4. | no.93-02 新聞語料庫詞頻統計表 | 18 | 30 | 24 | _____ | _____ |
| 5. | no.93-03 新聞常用動詞詞頻與分類 | 10 | 15 | 13 | _____ | _____ |
| 6. | no.93-05 中文詞類分析 | 10 | 15 | 13 | _____ | _____ |
| 7. | no.93-06 現代漢語中的法相詞 | 5 | 10 | 8 | _____ | _____ |
| 8. | no.94-01 中文書面語頻率詞典（新聞語料詞頻統計） | 18 | 30 | 24 | _____ | _____ |
| 9. | no.94-02 古漢語字頻表 | 11 | 16 | 14 | _____ | _____ |
| 10. | no.95-01 注音檢索現代漢語字頻表 | 8 | 13 | 10 | _____ | _____ |
| 11. | no.95-02/98-04 中央研究院平衡語料庫的內容與說明 | 3 | 8 | 6 | _____ | _____ |
| 12. | no.95-03 訊息為本的格位語法與其剖析方法 | 3 | 8 | 6 | _____ | _____ |
| 13. | no.96-01 「搜」文解字－中文詞界研究與資訊用分詞標準 | 8 | 13 | 11 | _____ | _____ |
| 14. | no.97-01 古漢語詞頻表（甲） | 19 | 31 | 25 | _____ | _____ |
| 15. | no.97-02 論語詞頻表 | 9 | 14 | 12 | _____ | _____ |
| 16. | no.98-01 詞頻詞典 | 18 | 30 | 26 | _____ | _____ |
| 17. | no.98-02 Accumulated Word Frequency in CKIP Corpus | 15 | 25 | 21 | _____ | _____ |
| 18. | no.98-03 自然語言處理及計算語言學相關術語中英對譯表 | 4 | 9 | 7 | _____ | _____ |
| 19. | no.02-01 現代漢語口語對話語料庫標註系統說明 | 8 | 13 | 11 | _____ | _____ |
| 20. | Computational Linguistics & Chinese Languages Processing (One year)<br>(Back issues of *IJCLCLP*: US$ 20 per copy) | --- | 100 | 100 | _____ | _____ |
| 21. | Readings in Chinese Language Processing | 25 | 25 | 21 | _____ | _____ |
| | | | | **TOTAL** | _____ | _____ |

**10% member discount: ___________Total Due:__________**

- **OVERSEAS USE ONLY**
- PAYMENT： □ Credit Card ( Preferred )  
  □ Money Order or Check payable to "The Association for Computation Linguistics and Chinese Language Processing " or “中華民國計算語言學學會”
- E-mail：aclclp@hp.iis.sinica.edu.tw

Name (please print): ____________________________ Signature: ____________________________

Fax: __________________________________________ E-mail: ______________________________

Address：____________________________________________________________________________

# 中華民國計算語言學學會

## 相關出版品價格表及訂購單

| 編號 | 書目 | 會　員 | 非會員 | 冊數 | 金額 |
|---|---|---|---|---|---|
| 1. | no.92-01, no. 92-04 (合訂本)　ICG 中的論旨角色 與 A conceptual Structure for Parsing Mandarin--its Frame and General Applications-- | NT$ 80 | NT$ 100 | _____ | _____ |
| 2. | no.92-02, no. 92-03 (合訂本) V-N 複合名詞討論篇 與V-R 複合動詞討論篇 | 120 | 150 | _____ | _____ |
| 3. | no.93-01　新聞語料庫字頻統計表 | 120 | 130 | _____ | _____ |
| 4. | no.93-02　新聞語料庫詞頻統計表 | 360 | 400 | _____ | _____ |
| 5. | no.93-03　新聞常用動詞詞頻與分類 | 180 | 200 | _____ | _____ |
| 6. | no.93-05　中文詞類分析 | 185 | 205 | _____ | _____ |
| 7. | no.93-06　現代漢語中的法相詞 | 40 | 50 | _____ | _____ |
| 8. | no.94-01　中文書面語頻率詞典（新聞語料詞頻統計） | 380 | 450 | _____ | _____ |
| 9. | no.94-02　古漢語字頻表 | 180 | 200 | _____ | _____ |
| 10. | no.95-01　注音檢索現代漢語字頻表 | 75 | 85 | _____ | _____ |
| 11. | no.95-02/98-04　中央研究院平衡語料庫的內容與說明 | 75 | 85 | _____ | _____ |
| 12. | no.95-03　訊息為本的格位語法與其剖析方法 | 75 | 80 | _____ | _____ |
| 13. | no.96-01 「搜」文解字－中文詞界研究與資訊用分詞標準 | 110 | 120 | _____ | _____ |
| 14. | no.97-01　古漢語詞頻表（甲） | 400 | 450 | _____ | _____ |
| 15. | no.97-02　論語詞頻表 | 90 | 100 | _____ | _____ |
| 16 | no.98-01　詞頻詞典 | 395 | 440 | _____ | _____ |
| 17. | no.98-02　Accumulated Word Frequency in CKIP Corpus | 340 | 380 | _____ | _____ |
| 18. | no.98-03 自然語言處理及計算語言學相關術語中英對譯表 | 90 | 100 | _____ | _____ |
| 19. | no.02-01 現代漢語口語對話語料庫標註系統說明 | 75 | 85 | _____ | _____ |
| 20 | 論文集 COLING 2002 紙本 | 100 | 200 | _____ | _____ |
| 21. | 論文集 COLING 2002 光碟片 | 300 | 400 | _____ | _____ |
| 22. | 論文集 COLING 2002 Workshop 光碟片 | 300 | 400 | _____ | _____ |
| 23. | 論文集 ISCSLP 2002 光碟片 | 300 | 400 | _____ | _____ |
| 24. | 交談系統暨語境分析研討會講義<br>（中華民國計算語言學學會1997第四季學術活動） | 130 | 150 | _____ | _____ |
| 25. | 中文計算語言學期刊（一年四期）　年份：______<br>（過期期刊每本售價500元） | --- | 2,500 | _____ | _____ |
| 26. | Readings of Chinese Language Processing | 675 | 675 | _____ | _____ |
| 27. | 剖析策略與機器翻譯 1990 | 150 | 165 | _____ | _____ |
|  |  |  | 合　計 | _____ | _____ |

※　此價格表僅限國內 (台灣地區) 使用

劃撥帳戶：中華民國計算語言學學會　　劃撥帳號：19166251

聯絡電話：(02) 2788-3799 轉1502

聯絡人：　黃琪 小姐、何婉如 小姐　　E-mail:aclclp@hp.iis.sinica.edu.tw

訂購者：______________________　收據抬頭：______________________

地　　址：____________________________________________

電　　話：______________________　E-mail: ______________________

# Information for Authors

**International Journal of Computational Linguistics and Chinese Language Processing** (IJCLCLP) invites submission of original research papers in the area of computational linguistics and speech/text processing of natural language. All papers must be written in English or Chinese. Manuscripts submitted must be previously unpublished and cannot be under consideration elsewhere. Submissions should report significant new research results in computational linguistics, speech and language processing or new system implementation involving significant theoretical and/or technological innovation. The submitted papers are divided into the categories of regular papers, short paper, and survey papers. Regular papers are expected to explore a research topic in full details. Short papers can focus on a smaller research issue. And survey papers should cover emerging research trends and have a tutorial or review nature of sufficiently large interest to the Journal audience. There is no strict length limitation on the regular and survey papers. But it is suggested that the manuscript should not exceed 40 double-spaced A4 pages. In contrast, short papers are restricted to no more than 20 double-spaced A4 pages. All contributions will be anonymously reviewed by at least two reviewers.

**Copyright**：It is the author's responsibility to obtain written permission from both author and publisher to reproduce material which has appeared in another publication. Copies of this permission must also be enclosed with the manuscript. It is the policy of the CLCLP society to own the copyright to all its publications in order to facilitate the appropriate reuse and sharing of their academic content. A signed copy of the IJCLCLP copyright form, which transfers copyright from the authors (or their employers, if they hold the copyright) to the CLCLP society, will be required before the manuscript can be accepted for publication. The papers published by IJCLCLP will be also accessed online via the IJCLCLP official website and the contracted electronic database services.

**Style for Manuscripts:** The paper should conform to the following instructions.
***1. Typescript:*** Manuscript should be typed double-spaced on standard A4 (or letter-size) white paper using size of 11 points or larger.
***2. Title and Author:*** The first page of the manuscript should consist of the title, the authors' names and institutional affiliations, the abstract, and the corresponding author's address, telephone and fax numbers, and e-mail address. The title of the paper should use normal capitalization. Capitalize only the first words and such other words as the orthography of the language requires beginning with a capital letter. The author's name should appear below the title.
***3. Abstracts and keywords:*** An informative abstract of not more than 250 words, together with 4 to 6 keywords is required. The abstract should not only indicate the scope of the paper but should also summarize the author's conclusions.
***4. Headings:*** Headings for sections should be numbered in Arabic numerals (i.e. 1.,2....) and start form the left-hand margin. Headings for subsections should also be numbered in Arabic numerals (i.e. 1.1. 1.2...).
***5. Footnotes:*** The footnote reference number should be kept to a minimum and indicated in the text with superscript numbers. Footnotes may appear at the end of manuscript
***6. Equations and Mathematical Formulas:*** All equations and mathematical formulas should be typewritten or written clearly in ink. Equations should be numbered serially on the right-hand side by Arabic numerals in parentheses.
***7. References:*** All the citations and references should follow the APA format. The basic form for a reference looks like

Authora, A. A., Authorb, B. B., & Authorc, C. C. (Year). Title of article. *Title of Periodical, volume number*(issue number), pages.

Here shows an example.

Scruton, R. (1996). The eclipse of listening. *The New Criterion, 15*(30), 5-13.

The basic form for a citation looks like (Authora, Authorb, and Authorc, Year). Here shows an example. (Scruton, 1996).
Please visit the following websites for details.
(1) APA Formatting and Style Guide (http://owl.english.purdue.edu/owl/resource/560/01/)
(2) APA Stytle (http://www.apastyle.org/)

**No page charges** are levied on authors or their institutions.

**Final Manuscripts Submission:** If a manuscript is accepted for publication, the author will be asked to supply final manuscript in MS Word or PDF files to clp@hp.iis.sinica.edu.tw

**Online Submission**: http://www.aclclp.org.tw/journal/submit.php

**Please visit the IJCLCLP Web page at http://www.aclclp.org.tw/journal/index.php**

International Journal of
Computational Linguistics & Chinese Language Processing
vol.19 no.1 March 2014

# Contents

**Papers**

章無疵也章之明靡句
無玷也句之清英字不
妄也文賦曰選義按部
考辭就班就所傳達者
觀之禮記曰發志為言
發言為名傳曰言以足志
文以足言易曰書不盡言
言不盡意詩序曰在心為
志發言為詩情動於中
而形於言蓋情志發而
語言成語言工而文字傳
也

**ISSN: 1027-376X**

The Association for Computational Linguistics and Chinese Language Processing