Meaning Representation and Meaning Instantiation for Chinese Nominals*

Kathleen Ahrens[1], Li-li Chang[2], Keh-Jiann Chen[2], Chu-Ren Huang[2]
[1]National Taiwan University
[2]Academia Sinica

Abstract

The goal of this paper is to explicate the nature of Chinese nominal semantics, and to create a paradigm for nominal semantics in general that will be useful for natural language processing purposes. We first point out that a lexical item may have two meanings simultaneously, and that current models of lexical semantic representation cannot handle this phenomena. We then propose a meaning representation that deals with this problem, and also discuss how the meanings involved are instantiated. In particular we posit that in addition to the traditional notion of sense differentiation, each sense may have different meaning facets. These meaning facets are linked to their sense or to other meaning facets through one of two ways: meronymic or metonymic extension.

1. Introduction

Lexical ambiguity resolution is a central concern of natural language processing (Small et al., 1988). The traditional way of looking at the problem is to list the various meanings that a word has, and write a rule-based program to pick the appropriate meaning for the context. Both Categorical Grammar and Montague Semantics, for example, assume that meanings are discrete and that there is a one-to-one correspondence between a lexical item and its meaning translation. The discrete meaning hypothesis provides the conceptual basis for most of the previous literature on ambiguity resolution and semantic resolution. In short, ambiguity resolution is viewed as trying to choose from several discrete meanings that share the same linguistic form (i.e. lexical form). While this approach can provide an algorithm to

* This paper is jointly authored. The names of the authors are listed in alphabetical order.

identify an appropriate meaning in a given context, it cannot account for novel uses of lexical items.

More recent work addresses this problem. Pustejovsky's (1995) Generative Lexicon provides a framework (i.e. qualia structure) for possible meanings, and discusses under what conditions which meaning will be chosen (i.e semantic coercion). His account is especially useful in dealing with the creative use of words in novel contexts, an area that had been previously ignored due to the assumption that either a) the novel usage could be listed if necessary, and b) often it was deemed not necessary to list these novel meanings because they occurred so rarely.

However, one issue that Pustejovsky and others have yet to account for is the fact that lexical meaning can be **actively complex**. All models of lexical ambiguity resolution assume that only one solution exists in a given context. In fact, what we will show is that more than one meaning can co-exist in the same context. A lexical item is actively complex if it allows simultaneous multiple interpretations. We will propose a meaning representation for lexical items that captures this complexity.

In addition, although Pustejovsky provides the framework to exclude the possible meanings, he cannot predict the relationship among the meanings, nor allow for cases where different meanings seem to exist simultaneously. Within the general theory of the Generative Lexicon, Copestake and Briscoe (1995) deal with meaning extension by either underspecification or lexical rules, which also implies that only one meaning can be expressed at any given time.

In our account, we will demonstrate that meaning can be predicted from its context by looking at a) the semantic class of the item, and b) its possible meaning extensions. Our account has the advantage of being able to account for a wider range of linguistic data, including puns and polysemous uses, in addition to novel extensions. Our account also has the advantage of being both computationally parsimonious, as well as conceptually intuitive.

Our paper is divided as follows: in section 2, we will first present background information and definitions concerning the different kinds of ways that meanings can vary. In section 3, we will present our arguments for the active complexity of lexical meaning, present a representation that can handle active complexity, and also give reasons for the conceptual intuitiveness of the model. In section 4, we will discuss the meaning extensions that have been found to date. Section 5 discusses the hierarchical information that is passed

from a semantic class to an individual item of that class. Section 6 summarizes our findings and suggests future areas of research.

2. Background

In this paper we devise a meaning representation for nominals (and Chinese nominals in particular) such that all meaning aspects of a noun are dealt with parsimoniously. Nouns, at first glance, do not seem to warrant representational complexity. When one is asked to think of a noun, one commonly thinks of a concrete object, such as 'paper'. When asked to define it, one could reply that it is a thin, white, rectangular object (appearance) made from the pulp of trees (origin) that people nowadays use to write and print on (function). But 'paper', even if we do not talk about its additional meanings in compound items such as 'wrapping paper', 'tissue paper' etc., has a variety of meanings including: a piece of paper, a newspaper, the office where a newspaper is written, and an academic paper. This phenomenon is not language specific. For example, in Mandarin Chinese, the word ' magazine' can refer to the physical object (1a), or the information contained within (1b), or the publishing house (1c).

(1a) 他 手 上 拿 了 本 雜誌。
ta shou shang na le ben zazhi
he hand on hold asp. CL magazine
'He is holding one magazine in his hand.'

(1b) 我們 從 雜誌 中 得到 許多 寶貴的 資料。
women cong zazhi zhong dedao xuduo baoguide ziliao
we from magazine within obtain many precious data
'We have obtained a lot of precious data from magazines.'

(1c) 美國 各 大 雜誌 無不 挖空 心思 爭取 採訪 機會。
meiguo ge da zazhi wubu wakong xinsi zhengqu caifang jihui
America every big magazine do dig-empty mind fight-for interview chance
'Major American magazines fight for interview opportunities.'

Nor is this phenomena limited to words relating to items that may contain information such as papers and magazines. The word 'tian' in Chinese can refer to the sky (2a), God (2b), weather (2c), time (2d), day(s) (2e), or nature (2f).

(2a) 抬頭　　望　　著　湛藍的　　天。
taitou　　wang　zhe　zhanlande　tian
raise head　watch　asp.　blue　　sky
'Raise one's head and look at the blue sky.' ('Tian' refers to sky.)

(2b) 中國人　　說　福　　自　　天　來。
zhongguoren　shuo　fu　　zi　　tian　lai
Chinese　　say　happiness　from　sky　come
'Chinese say, *happiness comes from heaven.*' ('Tian' refers to God/heaven.)

(2c) 天　冷　時　別　忘　了　加　件　衣服。
tian　leng　shi　bie　wang　le　jia　jian　yifu
sky　cold　time　not　forget　asp.　add　CL　clothes
'Don't forget to put on more clothes when the weather is cold.'
('Tian' refers to weather.)

(2d) 天　不　早　了。
tian　bu　zau　le
sky　not　early　particle
'It is not early.' ('Tian' refers to time.)

(2e) 他　在　這裡　待　了　一　整　天。
ta　zai　zheli　dai　le　yi　zheng　tian
he　in　here　stay　asp.　one　whole　sky/day
'He has stayed here for one whole day.' ('Tian' refers to day(s).)

(2f) 人類　　是　大部分　動物　的　天敵。
renlei　　shi　dabufen　dongwu　de　tiandi
human being　is　most　animal　's　natural enemy
'Human beings are the natural enemy of almost all animals.' ('Tian' refers to nature.)

The examples we have given above are all examples of polysemy, which is when a word has several, related meanings. But meanings can also be unrelated, as in the case of the two meanings for 'bank' (i.e. 'financial institution' and 'land on the side of a river'). A noun that has two unrelated meanings is referred to as homonymous. Meanings for a word can also be vague or underspecified. An example of this in English is 'aunt' which can refer to someone's parent's sister, where the gender as to the parent is unspecified. (The parent's gender in other languages, such as Mandarin, is important and specified.) The difference as to whether a word is ambiguous or polysemous depends on the perceived relationship (or lack thereof) between the meanings. The distinction between vagueness and polysemy ' involves the question whether a particular piece of semantic information is part of the

underlying semantic structure of the item, or is the result of a contextual (and hence pragmatic) specification' (Geerarts 1993:228).

This definition, however, cannot be applied as straightforwardly as it appears. Consider example (1) above. It could be the case that there is no underlying semantic structure for the three meanings (that is, they are vague), and that context alone 'brings out' these meanings. But 1) intuitively these meanings seem to have an underlying structure, and 2) nouns of a similar semantic class (i.e. magazine) have similar meanings, which indicates that an underlying structure exists. If it is the case that the pieces of semantic information are part of the underlying structure of the item, then we must deal with paradoxical situation (given the definition above) that these different meanings are brought out in different contexts.

Tuggy (1993) points out that ambiguity, polysemy and vagueness are better dealt with on a continuum, rather than as sets with discrete boundaries. The prototypical case of ambiguity is where well-entrenched and salient semantic structures are associated with the same phonological representation, and there is no clear subsuming semantic schema. The prototypical case of vagueness is where the meanings are not well-entrenched, and there is a clear subsuming semantic schema (as in the case of parent's sister for 'aunt'). Polysemy is viewed as being in between these two extremes, with there are well-entrenched and salient semantic structures associated with the same phonological representation, but there is also a subsuming schema.

3. Meaning Representation

3.1 Active Complexity of Lexical Items

The above discussion has assumed that one meaning is chosen in a given context. But that is not necessarily the case. There are two types of active complexity in natural language. The first is 'triggered complexity' and involves puns. For example, in (3) either *liquor* and *shipyard* is possible as the meaning of *port*, but it is also possible for both meanings to exist at the same time.

(3) After the accident, the captain went straight for the *port*.

Example (3) can mean that a) the captain went straight for shore (but humorously implies that the captain was so shook up as to need a drink), or b) that he went straight for his bottle of liquor and also towards the shore (although this is much less likely since this interpretation is not seen as humorous.

The phenomena in example (3) is a pun. Puns are a humorous play on ambiguous words. Because puns are used for special linguistic purposes (such as humor), and because it is the effect of co-existing meanings that creates the humor, this phenomena has not previously been considered to be relevant to lexical semantic analysis and lexical representation. The complexity is triggered since it must be initiated by the speaker.

Second, in Chinese, nouns can be actively complex, even when there is no pun or vagueness intended. This is 'latent complexity.' In (4), for example 'book' *must* be understood as both a physical object, and as information.

(4) 張三 在 翻閱 那 一 本 書。
Zhangsan zai fanyue na yi ben shu.
Zhangsan PROG turn page/read that one CL book
'Zhangsan is turning the pages of the book and reading it.'

In fact, such latent complexity also exists in English nominal semantics. It is well-known that words referring to building apertures, such as door or window are often lexically ambiguous with the structure built to block that aperture. Thus, *door* in 'the door is heavy' could only refer to the structure, while *door* in 'John just walked in the door' can only refer to the aperture. However, in the sentence, 門很寬 men hen kuan 'The door is wide', both the aperture and structure's meanings exist simultaneously. We think this kind of data presents the strongest argument *against* representing nominal semantics as discrete meaning translations, and for representing nominal semantics as structured meanings connected by conceptual links, such as the qualia structure in Pustejovsky's Generative Semantics. However, since we have shown that different but related meanings can coexist in the same context, Pustejovsky's formulation where related meanings are represented as different attribute value pairs in a feature matrix is inadequate since only one attribute value pair can be picked in each context. We posit that these related meanings are like the facets of a three-dimensional object, such as a diamond, where the meaning instantiation could be a straightforward single facet or multiple connected facets, depending on the contexts.

3.2 Meaning Representation

The meaning representation that we select is quite straightforward, but differs from other representations in several crucial respects. First, words are listed (following Chinese lexicographic tradition) in terms of their orthographic representation. Then the senses for each word are listed. The phonological representations are associated with each sense listing, and may or may not be the same. Second, the sense differentiation includes senses that are related (polysemous senses) as well as unrelated (homonymous senses). There is no attempt in this representation to distinguish clearly between those meanings that are polysemous or homonymous. This is because speakers tend to draw their own conclusions about the relationships between senses (i.e. many speakers see a relationship between '*ear* of corn' and '*ear* that you hear with', although there is no historical or semantic relationship whatsoever (Lyons 1977)). However, if a study was run on native speakers to find out their understanding of the relative closeness of relationship among meanings, this information could be incorporated into our representation by simply indicating which senses should be grouped together. Third, and most importantly, our lexical representation has **meaning facets** located within each sense. Meaning facets reflect an aspect of a sense. For example, in (5) we show an example of a word with one sense, which has different meaning facets.

(5) 雜誌 — **Sense$_1$:** ZAZHI ***magazine*** -- meaning facet$_1$: *physical object*
-- meaning facet$_2$: *information contained within*
-- meaning facet$_3$: *institution that publishes magazine*

In (6) we give an example of a word with four different senses, of which one has three different meaning facets.

(6) 天 — **Sense$_1$:** TIAN ***sky*** -- meaning facet$_1$: *sky as a physical object (that can be viewed)*
-- meaning facet$_2$: *God/heaven*
-- meaning facet$_3$: *weather*
— **Sense$_2$:** TIAN ***time***
— **Sense$_3$:** TIAN ***day***
— **Sense$_4$:**TIAN ***nature***

How do we decide whether a certain meaning is a sense or a meaning facet? A meaning facet is an extension from a particular sense. It has the following three properties: 1) it can appear in the same context as other meaning facets, 2) it is an extension from a core sense or from another meaning facet (unless it is the core sense), 3) nouns of the same semantic classes will have similar sense extensions to related meaning facets. Individual senses, on the other hand, 1) cannot appear in the same context (unless the complexity is triggered), 2) have no core sense from which it is extended, or it is very hard to concisely define what the core sense would be, and 3) no logical/conceptual links can be established between the two senses.

For example, in (7) below, we can see that the meaning of sky (as a physical object) and God can appear in the same context, as can sky (as a physical object) and weather (8), sky (as a physical object), God, and weather (9). Thus, they are all different meaning facets of the first sense in (6).

(7) 有 人 開始 不 敬 天 也 不 拜 天 了。
you ren kaishi bu jing tian ye bu bai tian le
there're person begin not respect sky and not worship sky particle
'There are people who ceased to respect heaven or to worship heaven.'
('Tian' refers to both sky and God/heaven.)

(8) 天 放晴 了。
tian fangqing le
sky become sunny particle
'It became sunny.' ('Tian' refers to both sky and weather.)

(9) 農民 長久 靠 天 依 地 的 生活。
nongmin changjiou kau tian yi di de shenghuo
farmer long depend sky depend ground DE live
'Farmers have long lived a life that depends on heaven and earth.'
('Tian' refers to sky, God, and weather.)

The above examples also demonstrate that only one *sense* can occur in any given context. The sense of 'time' or 'day' or 'nature' is not available in any one of the above

contexts.[1] Only meaning facets of a particular sense can be available in the same context. Context, in effect, selects which sense is made available. Context may also select a particular meaning facet, as in (2a)- (2c), but it does not necessarily have to, because context may activate several meaning facets at once, as in (7) - (9).

What aspects of context help to pick a sense or a meaning facet? Verbs and prepositions are usually instrumental in determining which meaning can occur in which context. For example, in the above instance, the meaning of 'God' can only occur with volitional verbs and cannot occur with verbs having to do with pure locative. The type of contextual information that picks out one sense or one meaning facet is an important area of future research.

3.3 Conceptual Advantages

Viewed from this perspective, context *always* plays a role in determining which meaning is chosen, whether the word is ambiguous, polysemous, or vague. Tuggy's meaning models were two dimensional. But we suggest that a 3-dimension model allows for a greater understanding of the relationship between meaning and context. Imagine a multi-faceted object, such as a cube. Imagine that there is a core in the center of the cube, and that there are lines that radiate out to each of the six surfaces (i.e. this would be the case for a word that had six senses). The core represents the orthographic representation of the word, and each surface represents a different sense of the word and its associated phonological representation (i.e. the information that is bolded in our lexical representation above). Furthermore, from each surface of the cube, there may also be (dotted) lines that radiate out to additional surfaces, which are the facets of that particular sense (i.e. the non-bolded information in our lexical representation above). Thus, when context turns the cube so that one particular sense surface is shown to a light source (i.e. the hearer) then light is reflected from only that surface, and only that sense is computed. In the case, however, where context

---

[1] 'Time' might be viewed as a meaning facet of the sense 'sky', as shown by the identical strings in (i) and (ii).

(i) [S天 [VP黑 了]] 。
tian hei le
sky dark particle
'The sky turned dark.'

(ii) [S[VP[V天 黑] 了]]] 。
tianhei le
sunset particle
'The sun has set (i.e. it is late).'

However, the interpretation in (i) is a subject-predicate sentence, while the interpretation in (ii) involves a disyllabic lexical item. Thus, these two sentences are structurally different and no latent complexity is

turns the cube so that a sense surface that has meaning facets extending from it is shown to a light source, the light can reflect off of any one, or any combination of the meaning facets, just as light can reflect from the different facets of a diamond. Our representation, then, is not only computationally adequate, it is also conceptually intuitive.

In what follows we present the types of links that can occur in noun meaning representations, and we also present the underlying schema for the information contained in each meaning facet.

4. Meaning Links

In our model the meaning representation is structured, and the structure is built upon meaning links. One implication of this model is that a semantic class will inherit both traditional semantic features as well meaning link structures. Lexical semantic issues will therefore be defined in terms of types of possible meaning links and constraints on meaning extensions through these links.

The relationship between a sense and its meaning facets is an area that deserves in depth research and analysis (Ahrens et al., In prep.). What follows is a preliminary report of our findings to date. We have found that there are two main ways that meaning facets can extend either from a sense or from another meaning facet: meronymic and metonymic extensions.

4.1 Meronymic extensions

Meronymic extensions involve both the whole standing for part, and part standing for whole. We observe that meronymic extensions are driven by cognitive and conceptual saliency. For example, in 那把刀子很利 na ba daozi hen li (The knife is sharp), *knife* actually refers to the blade of the knife. This meronymic extension is motivated by the fact that 'blade' is the locus of cutting, the most salient function of knife. We also observe that such cognitively driven extensions are not sensitive to blocking effects. For instance, the instance of the specific term 'blade' does not block us from saying 'the knife is sharp.' Our speculation here is that only conventionalized usages are subject to blocking effects since blocking is the result of conventionalization.

---

involved.

In the case of part standing for whole, cognitive saliency is again the prime motivator of the extension. For example, in the case of 院子裡有許多梅花 yuanzi li you xuduo meihua (there are many plum-flowers in the garden), *plum-flowers* stands for the whole plum tree. The plum flower with its color and scent and endurance in cold weather is the most cognitively salient aspect of the plum tree (for Chinese).

4.2 Metonymic Extensions

Metonymic extensions are different from meronymic extensions in that the extended meaning is related to the origin of the basic sense, but is not inherent to the basic sense (cf. the part-whole relation above). Metonymic extensions are typically driven by certain eventive relationships such as the ones encoded in Pustejovsky's qualia structure. Unlike meronymic extensions, metonymic extensions are often sensitive to blocking effects. For instance, the grinding extension allows the individual terms to refer to a mass produced from that individual. For example, '一盤白菜 yi pan baicai' (a dish of cabbage), the basic meaning '白菜 baicai' refers to the cabbage plant, but after the grinding extension it refers to a mass noun. But in the case of rice '米 mi', the grinding extension does not work, because there is a term '飯 fan' (cooked rice) already.

4.3 Partial list of Meaning Links

We give here a partial list of the meaning links found to date. We also provide the list of semantic classes that we have found to inherit these links.

I. Meronymic Extensions
- 1. Whole for part
  - a. whole → functional part {semantic class: artifacts, buildings}
  - b. whole → sentiently salient part {semantic class: body parts}
- 2. Part for whole
  - a. conceptually salient part → whole {semantic class: fruit, flower}

II. Metonymic Extensions

1. agentivization
   a. information media → information creator {semantic class: publications}
2. product instantiation
   a. institution → product {semantic class: manufacturer, trademarks}
3. grinding
   a. individual → mass {semantic class: vegetables, fruits}
4. portioning
   a. information media → information {semantic class: publications}
   b. container → containee
   c. body part → function
5. space mark-up
   a. landmark → space in vicinity {semantic class: locations, landmarks}
   b. structure → aperture {semantic class: doors, windows}
   c. institution → locus {semantic class: institutions}
6. time mark-up
   a. event → temporal period
   b. object → process
   c. locus → duration

We have found that these two types of links are the most productive among meaning extensions. This might be because these types of extensions refer only to the knowledge concerning the lexical item itself. Metaphorical extensions, on the other hand, map a domain of knowledge that does not have anything to do with the lexical item onto the domain of knowledge surrounding the lexical item. Thus, metaphorical extensions are clearly conceptually more complex than metonymic and meronymic extensions, and will be the focus of future research.

5. Meaning Inheritance

Another important issue in lexical semantics is the semantic class. Traditionally, the taxonomic hierarchies are discussed in terms of ISA relationship and inherited features, such as humanness and animacy (Chen and Cha 1988, Sowa 1993). However, this simplistic traditional model (such as the Schank's well-known semantic network) have difficulties when certain nodes do not necessarily inherit all the features from the higher nodes. For example, an ostrich is a bird, but it cannot inherit the feature of [+flight] because it does not fly. Default override of inheritance is computationally plausible though costly.

The other problem with traditional semantic hierarchies has to do with multiple inheritance. For instance, it is intuitive to classifiy '籃球 lan-qiu' (basketball) as a physical

object. However, it is also clearly an abstract event (i.e. the basketball game). Hence there is cross-taxonomic paradox, which is usually accounted for with the computationally costly mechanism of multiple inheritance (Briscoe et al., 1993).

In our model, both kinds of inheritance problems disappear since what a semantic class shares is a partial structure of semantic links. That is, we will annotate meaning links to a semantic class, and these links will be inherited by all the members of the class.[2] In the case of '球 qiu' (ball), it inherits the metonymic link of a round physical object and extends to the game play with the object. This explanation is more parsimonious since it reduces the costly computation of multiple inheritance and makes most cases of the local overriding of inheritance unnecessary. It is also conceptually powerful in allowing richer semantic representation. For instance, the semantic class of flowers will inherit the meronymic extension of part for whole.

6. Conclusion: Implementation and Implications

Traditional methods of dealing with ambiguity and vagueness in natural language processing have been complicated by the on-line compilations that are usually necessary to deal with the 'additional' meanings created by the context. But our account postulates multiple senses and structured ways of linking additional meaning facets to the senses so that the information is all listed in the representation, and therefore easier to access. Our proposal is to have not only the different senses of a word listed, but also its different meaning facets. We claim that there will be conceptual or logical relationships between the facets and their senses.

For example, the meaning links between the different facets of 'zazhi' (magazine) are as follows: the first meaning link refers to the concept of magazine as a physical object, the second meaning link is a metonymic extension that relates media to information, and the third meaning link is a metonymic extension that relates information media to information creator (c.f. Section 4.3). The organization that we have proposed here is a shallow structure, with only two levels: the sense level and the meaning facet level. Both levels can be annotated with meaning links. Conceptually it is as explanatory as a theory where all the

[2] Of course, the lexicon would have to specify any blocking effects where the linking does not apply.

meaning links are structurally represented. This is because all represented meaning links can be traced, and a (semantic-class-based) meaning derivation tree can be established off-line. Moreover, not having an overt tree of meaning extensions allows us to avoid multiple-inheritance and blocking problems. A shallow structure also allows efficient access, reflecting the psychological reality that the depth of meaning derviation is not relevant in lexical access.

In this paper we have proposed a meaning representation for Chinese nominal semantics, as well as a paradigm for nominal semantics in general that will be useful for natural language processing purposes. We pointed out that a lexical item may have two meanings simultaneously, and moreover that current models of lexical semantic representation cannot handle this phenomena. We then proposed a meaning representation to account for this phenomena, and also discussed how the meanings involved are instantiated. We postulate that in addition to the traditional notion of sense differentiation, each sense may have different meaning facets. These meaning facets are linked to their sense or to other meaning facets through one of two ways: meronymic or metonymic extension. We also point out that instead of a traditional taxonomic relationship, what is being inherited in addition to semantic features is meaning extensions/relations, such that words of the same semantic class have same meaning extensions.

The representation proposed here is the result of extensive corpus-based studies of the 40 most productive nominal endings in Mandarin (CKIP 1995). These productive nominal endings in turn each derive scores of highly frequent nouns. Hence we have accounted for a substantial portion of Chinese noun usages. We have also provided detailed semantic representation of the nominal heads based on our proposed representation. This is a significant first step towards the comprehensive formal representation of Mandarin nominal semantics and is also the first step towards fully automated Mandarin Language Understanding.

References

Ahrens, K., L.-L. Chang, K.-J. Chen, and C.R. Huang, "Meronymic and Metonymic Extensions: A Unified Account of Multiple Meanings," In preparation.

Briscoe, E., J. Copestake, and V. de Paiva, Inheritance, Defaults and the Lexicon. Cambridge University Press, 1993.

Chen, K.-J., and C.-S. Cha, "The Design of a Conceptual Structure and Its Relation to the Parsing of Chinese Sentences," Proceedings of 1988 International Conference on Computer Processing of Chinese and Oriental Languages (ICCPCOL), 1988, pp.428-431.

CKIP, "Contents and Explanations of Sinica Corpus," CKIP Technical Report. 95-02. Nankang: Academia Sinica, 1995.

Copestake, A., and T. Briscoe, "Semi-productive Polysemy and Sense Extension," Journal of Semantics, 12, 1995, pp.15-67.

Geerarts, D., "Vagueness's puzzles, polysemy's vagaries," Cognitive Linguistics, 4.3, 1993, 223-272.

Lyons, J., Semantics, Cambridge University Press, 1977.

Pustejovsky, J., Generative Lexicon, MIT Press, 1995.

Small, S., G. Cottrell, and M. Tanenhaus, Lexical Ambiguity Resolution: Perspectives from Psycholinguistics, Neuropsychology and Artificial Intelligence, Morgan Kaufmann Publishers, 1988.

Sowa, J, "Lexical Structure and Conceptual Structure," in Semantics and the Lexicon, Pustejovsky (Ed.), Kluwer Academic Publishers, 1993, pp.223-262.

Tuggy, D., "Ambiguity, Polysemy and Vagueness," Cognitive Linguistics. 4.3, 1993, pp.273-290.

Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy

Jay J. Jiang
Department of Management Sciences
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
jjiang@uwaterloo.ca

David W. Conrath
MGD School of Business
McMaster University
Hamilton, Ontario, Canada L8S 4M4
conrathd@mcmaster.ca

Abstract

This paper presents a new approach for measuring semantic similarity/distance between words and concepts. It combines a lexical taxonomy structure with corpus statistical information so that the semantic distance between nodes in the semantic space constructed by the taxonomy can be better quantified with the computational evidence derived from a distributional analysis of corpus data. Specifically, the proposed measure is a combined approach that inherits the edge-based approach of the edge counting scheme, which is then enhanced by the node-based approach of the information content calculation. When tested on a common data set of word pair similarity ratings, the proposed approach outperforms other computational models. It gives the highest correlation value (r = 0.828) with a benchmark based on human similarity judgements, whereas an upper bound (r = 0.885) is observed when human subjects replicate the same task.

1. Introduction

The characteristics of polysemy and synonymy that exist in words of natural language have always been a challenge in the fields of Natural Language Processing (NLP) and Information Retrieval (IR). In many cases, humans have little difficulty in determining the intended meaning of an ambiguous word, while it is extremely difficult to replicate this process computationally. For many tasks in psycholinguistics and NLP, a job is often decomposed to the requirement of resolving the semantic relation between words or concepts. One needs to come up with a consistent computational model to assess this type of relation. When a word level semantic relation requires exploration, there are many potential types of relations that can be considered: hierarchical (e.g. IS-A or hypernym-hyponym, part-whole, etc.), associative (e.g. cause-effect), equivalence (synonymy), etc. Among these, the hierarchical relation represents the major and most important type, and has been widely studied and applied as it maps well to the human cognitive view of classification (i.e. taxonomy). The IS-A relation, in particular, is a typical representative of the hierarchical relation. It has been suggested and employed to study a special case of semantic relations — semantic similarity or semantic distance (Rada et al. 1989). In this study of semantic similarity, we will take this view, although it excludes some potential useful information that could be derived from other relations.

The study of words/terms relationships can be viewed in terms of the information sources used. The least information used are knowledge-free approaches that rely exclusively on the

corpus data themselves. Under the corpus-based approach, word relationships are often derived from their co-occurrence distribution in a corpus (Church and Hanks 1989, Hindle 1990, Grefenstette 1992). With the introduction of machine readable dictionaries, lexicons, thesauri, and taxonomies, these manually built pseudo-knowledge bases provide a natural framework for organising words or concepts into a semantic space. Kozima and Furugori (1993) measured word distance by adaptive scaling of a vector space generated from LDOCE (*Longman Dictionary of Contemporary English*). Morris and Hirst (1991) used Roget's thesaurus to detect word semantic relationships. With the recently developed lexical taxonomy WordNet (Miller 1990, Miller et al. 1990), many researches have taken the advantage of this broad-coverage taxonomy to study word/concept relationships (Resnik 1995, Richardson and Smeaton 1995).

In this paper, we will discuss the use of the corpus-based method in conjunction with lexical taxonomies to calculate semantic similarity between words/concepts. In the next section we will describe the thread and major methods in modelling semantic similarity. Based on the discussion, we will present a new similarity measure, which is a combined approach of previous methods. In section 3, experiments are conducted to evaluate various computational models compared against human similarity judgements. Finally, we discuss the related work and future direction of this study.

2. Semantic Similarity in a Taxonomy

There are certain advantages in the work of semantic association discovery by combining a taxonomy structure with corpus statistics. The incorporation of a manually built pseudo-knowledge base (e.g. thesaurus or taxonomy) may complement the statistical approach where "true" understanding of the text is unobtainable. By doing this, the statistics model can take advantage of a conceptual space structured by a hand-crafted taxonomy, while providing computational evidence from manoeuvring in the conceptual space via distributional analysis of corpora data. In other words, calculating the semantic association can be transformed to the estimation of the conceptual similarity (or distance) between nodes (words or concepts) in the conceptual space generated by the taxonomy. Ideally, this kind of knowledge base should be reasonably broad-coverage, well structured, and easily manipulated in order to derive desired associative or similarity information.

Since a taxonomy is often represented as a hierarchical structure, which can be seen as a special case of network structure, evaluating semantic similarity between nodes in the network can make use of the structural information embedded in the network. There are several ways to determine the conceptual similarity of two words in a hierarchical semantic network. Topographically, this can be categorised as node based and edge based approaches, which correspond to the information content approach and the conceptual distance approach, respectively.

2.1. Node-based (Information Content) Approach

One node based approach to determine the conceptual similarity is called the information content approach (Resnik 1992, 1995). Given a multidimensional space upon which a node represents a unique concept consisting of a certain amount of information, and an edge

represents a direct association between two concepts, the similarity between two concepts is the extent to which they share information in common. Considering this in a hierarchical concept/class space, this common information "carrier" can be identified as a specific concept node that subsumes both of the two in the hierarchy. More precisely, this super-class should be the first class upward in this hierarchy that subsumes both classes. The similarity value is defined as the information content value of this specific super-ordinate class. The value of the information content of a class is then obtained by estimating the probability of occurrence of this class in a large text corpus.

Following the notation in information theory, the information content (IC) of a concept/class c can be quantified as follows:

$$IC(c) = \log^{-1} P(c), \tag{1}$$

where $P(c)$ is the probability of encountering an instance of concept c. In the case of the hierarchical structure, where a concept in the hierarchy subsumes those lower in the hierarchy, this implies that $P(c)$ is monotonic as one moves up the hierarchy. As the node's probability increases, its information content or its informativeness decreases. If there is a unique top node in the hierarchy, then its probability is 1, hence its information content is 0.

Given the monotonic feature of the information content value, the similarity of two concepts can be formally defined as:

$$sim(c_1, c_2) = \max_{c \in Sup(c_1, c_2)} [IC(c)] = \max_{c \in Sup(c_1, c_2)} [-\log p(c)], \tag{2}$$

where $Sup(c_1, c_2)$ is the set of concepts that subsume both c_1 and c_2. To maximize the representativeness, the similarity value is the information content value of the node whose IC value is the largest among those super classes. In another word, this node is the "lowest upper bound" among those that subsume both c_1 and c_2.

In the case of multiple inheritances, where words can have more than one sense and hence multiple direct super classes, word similarity can be determined by the best similarity value among all the class pairs which their various senses belong to:

$$sim(w_1, w_2) = \max_{c_1 \in sen(w_1)} \max_{c_2 \in sen(w_2)} [sim(c_1, c_2)], \tag{3}$$

where $sen(w)$ denotes the set of possible senses for word w.

For the implementation of the information content model, there are some slightly different approaches toward calculating the concept/class probabilities in a corpus. Before giving the detailed calculation, we need to define two concept sets: *words(c)* and *classes(w)*. *Words(c)* is the set of words subsumed (directly or indirectly) by the class c. This can be seen as a sub-tree in the whole hierarchy, including the sub-tree root c. *Classes(w)* is defined as the classes in which the word w is contained; in another word, it is the set of possible senses that the word w has:

$$classes(w) = \{c | w \in words(c)\}. \quad (4)$$

Resnik (1995) defined a simple class/concept frequency formula:

$$freq(c) = \sum_{w \in words(c)} freq(w). \quad (5)$$

Richardson and Smeaton (1995) proposed a slightly different calculation by considering the number of word senses factor:

$$freq(c) = \sum_{w \in words(c)} \frac{freq(w)}{|classes(w)|} \quad (6)$$

Finally, the class/concept probability can be computed using maximum likelihood estimation (MLE):

$$P(C) = \frac{freq(c)}{N} \quad (7)$$

This methodology can be best illustrated by examples. Assume that we want to determine the similarities between the following classes: *(car, bicycle)* and *(car, fork)*. Figure 1 depicts the fragment of the WordNet (Version 1.5) noun hierarchy that contains these classes. The number in the bracket of a node indicates the corresponding information content value. From the figure we find that the similarity between *car* and *bicycle* is the information content value of the class *vehicle*, which has the maximum value among all the classes that subsume both of the two classes, i.e. *sim(car, bicycle)* = 8.30. In contrast, *sim(car, fork)* = 3.53. These results conform to our perception that cars and forks are less similar than cars and bicycles.

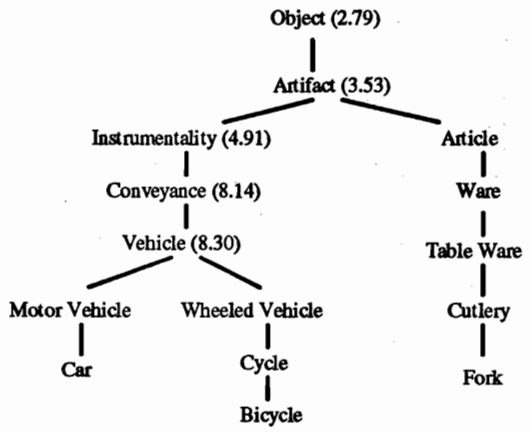


Figure 1. Fragments of the WordNet noun taxonomy

2.2. Edge-based (Distance) Approach

The edge based approach is a more natural and direct way of evaluating semantic similarity in a taxonomy. It estimates the distance (e.g. edge length) between nodes which correspond to

the concepts/classes being compared. Given the multidimensional concept space, the conceptual distance can conveniently be measured by the geometric distance between the nodes representing the concepts. Obviously, the shorter the path from one node to the other, the more similar they are.

For a hierarchical taxonomy, Rada et al. (1989) pointed out that the distance should satisfy the properties of a metric, namely: zero property, symmetric property, positive property, and triangular inequality. Furthermore, in an IS-A semantic network, the simplest form of determining the distance between two elemental concept nodes, A and B, is the shortest path that links A and B, *i.e.* the minimum number of edges that separate A and B (Rada et al. 1989).

In a more realistic scenario, the distances between any two adjacent nodes are not necessarily equal. It is therefore necessary to consider that the edge connecting the two nodes should be weighted. To determine the edge weight automatically, certain aspects should be considered in the implementation. Most of these are typically related to the structural characteristics of a hierarchical network. Some conceivable features are: local network density (the number of child links that span out from a parent node), depth of a node in the hierarchy, type of link, and finally, perhaps the most important of all, the strength of an edge link. We will briefly discuss the concept for each feature:

- With regard to network density, it can be observed that the densities in different parts of the hierarchy are higher than others. For example, in the *plant/flora* section of WordNet the hierarchy is very dense. One parent node can have up to several hundred child nodes. Since the overall semantic mass is of a certain amount for a given node (and its subordinates), the local density effect (Richardson and Smeaton 1995) would suggest that the greater the density, the closer the distance between the nodes (*i.e.* parent child nodes or sibling nodes).

- As for node depth, it can be argued that the distance shrinks as one descends the hierarchy, since differentiation is based on finer and finer details.

- Type of link can be viewed as the relation type between nodes. In many thesaurus networks the hyponym/hypernym (IS-A) link is the most common concern. Many edge-based models consider only the IS-A link hierarchy (Rada et al. 1989, Lee et al. 1993). In fact, other link types/relations, such as Meronym/Holonym (Part-of, Substance-of), should also be considered as they would have different effects in calculating the edge weight, provided that the data about the type of relation are available.

- To differentiate the weights of edges connecting a node and all its child nodes, one needs to consider the link strength of each specific child link. This could be measured by the closeness between a specific child node and its parent node, against those of its siblings. Obviously, various methods could be applied here. In particular, this is the place where corpus statistics could contribute. Ideally the method chosen should be both theoretical sound and computational efficient.

Two studies have been conducted in edge-based similarity determination by responding to the above concerns. Richardson and Smeaton (1995) considered the first two and the last factors in their edge weight calculation for each link type. Network density is simply counting the number of edges of that type. The link strength is a function of a node's information content value, and those of its siblings and parent nodes. The result of these two operations is then normalised by dividing them by the link depth. Notice that the precise formula of their implementation was not given in the paper.

Sussna (1993) considered the first three factors in the edge weight determination scheme. The weight between two nodes c_1 and c_2 is calculated as follows:

$$wt(c_1, c_2) = \frac{wt(c_1 \rightarrow_r c_2) + wt(c_2 \rightarrow_{r'} c_1)}{2d} \tag{8}$$

given

$$wt(x \rightarrow_r y) = \max_r - \frac{\max_r - \min_r}{n_r(x)} \tag{9}$$

where $\rightarrow_r$ is a relation of type r, $\rightarrow_{r'}$ is its reverse, d is the depth of the deeper one of the two, *max* and *min* are the maximum and minimum weights possible for a specific relation type r respectively, and $n_r(x)$ is the number of relations of type r leaving node x.

Applying this distance formula to a word sense disambiguation task, Sussna (1993) showed an improvement where multiple sense words have been disambiguated by finding the combination of senses from a set of contiguous terms which minimizes total pairwise distance between senses. He found that the performance is robust under a number of perturbations; however, depth factor scaling and restricting the type of link to a strictly hierarchical relation do noticeably impair performance.

In determining the overall edge based similarity, most methods just simply sum up all the edge weights along the shortest path. To convert the distance measure to a similarity measure, one may simply subtract the path length from the maximum possible path length (Resnik 1995):

$$sim(w_1, w_2) = 2d_{\max} - [\min_{c_1 \in sen(w_1)} \min_{c_2 \in sen(w_2)} len(c_1, c_2)], \tag{10}$$

where $d_{\max}$ is the maximum depth of the taxonomy, and the *len* function is the simple calculation of the shortest path length (*i.e.* weight = 1 for each edge).

2.3. Comparison of the Two Approaches

The two approaches target semantic similarity from quite different angles. The edge-based distance method is more intuitive, while the node-based information content approach is more theoretically sound. Both have inherent strength and weakness.

Rada et al. (1989) applied the distance method to a medical domain, and found that the distance function simulated well human assessments of conceptual distance. However,

Richardson and Smeaton (1995) had concerns that the measure was less accurate than expected when applied to a comparatively broad domain (e.g. WordNet taxonomy). They found that irregular densities of links between concepts result in unexpected conceptual distance outcomes. Also, without causing serious side effects elsewhere, the depth scaling factor does not adjust the overall measure well due to the general structure of the taxonomy (e.g. higher sections tend to be too similar to each other).

In addition, we feel that the distance measure is highly depended upon the subjectively pre-defined network hierarchy. Since the original purpose of the design of the WordNet was not for similarity computation purpose, some local network layer constructions may not be suitable for the direct distance manipulation.

The information content method requires less information on the detailed structure of a taxonomy. It is not sensitive to the problem of varying link types (Resnik 1995). However, it is still dependent on the skeleton structure of the taxonomy. Just because it ignores information on the structure it has its weaknesses. It normally generates a coarse result for the comparison of concepts. In particular, it does not differentiate the similarity values of any pair of concepts in a sub-hierarchy as long as their "smallest common denominator" (i.e. the lowest super-ordinate class) is the same. For example, given the concepts in Figure 1, the results of the similarity evaluation between *(bicycle, table ware)* and *(bicycle, fork)* would be the same. Also, other type of link relations information is overlooked here. Additionally, in the calculation of information content, polysemous words will have an exaggerated content value if only word (not its sense) frequency data are used (Richardson and Smeaton 1995).

2.4. A Combined Approach

We propose a combined model that is derived from the edge-based notion by adding the information content as a decision factor. We will consider various concerns of the edge weighting schemes discussed in the previous section. In particular, attention is given to the determination of the link strength of an edge that links a parent node to a child node.

We first consider the link strength factor. We argue that the strength of a child link is proportional to the conditional probability of encountering an instance of the child concept c_i given an instance of its parent concept p: $P(c_i \mid p)$.

$$P(c_i \mid p) = \frac{P(c_i \cap p)}{P(p)} = \frac{P(c_i)}{P(p)} \tag{11}$$

Notice that the definition and determination of the information content (see equations 1 and 5) indicate that c_i is a subset of p when a concept's informativeness is concerned. Following the standard argument of information theory, we define the link strength (LS) by taking the negative logarithm of the above probability. We obtain the following formula:

$$LS(c_i, p) = -\log(P(c_i \mid p) = IC(c_i) - IC(p). \tag{12}$$

This states that the link strength (LS) is simply the difference of the information content values between a child concept and its parent concept.

Considering other factors, such as local density, node depth, and link type, the overall edge weight (wt) for a child node c and its parent node p can be determined as follows:

$$wt(c,p) = \left(\beta + (1-\beta)\frac{\overline{E}}{E(p)}\right)\left(\frac{d(p)+1}{d(p)}\right)^{\alpha} [IC(c) - IC(p)]\, T(c,p), \qquad (13)$$

where $d(p)$ denotes the depth of the node p in the hierarchy, $E(p)$ the number of edges in the child links (i.e. local density), $\overline{E}$ the average density in the whole hierarchy, and $T(c,p)$ the link relation/type factor. The parameters α ($\alpha \geq 0$) and β ($0 \leq \beta \leq 1$) control the degree of how much the node depth and density factors contribute to the edge weighting computation. For instance, these contributions become less significant when α approaches 0 and β approaches 1.

The overall distance between two nodes would thus be the summation of edge weights along the shortest path linking two nodes.

$$Dist(w_1, w_2) = \sum_{c \in \{path(c_1,c_2) - LSuper(c_1,c_2)\}} wt(c, parent(c)) \qquad (14)$$

where $c_1=sen(w_1)$, $c_2=sen(w_2)$, and $path\ (c_1, c_2)$ is the set that contains all the nodes in the shortest path from c_1 to c_2. One of the elements of the set is $LSuper(c_1,c_2)$, which denotes the lowest super-ordinate of c_1 and c_2. In the special case when only link strength is considered in the weighting scheme of equation 13, *i.e.* $\alpha = 0$, $\beta = 1$, and $T(c,p) = 1$, the distance function can be simplified as follows:

$$Dist(w_1, w_2) = IC(c_1) + IC(c_2) - 2 \times IC(LSuper(c_1, c_2)) \qquad (15)$$

Imagine a special multidimensional semantic space where every node (concept) in the space lies on a specific axis and has a mass (based on its information content or informativeness). The semantic distance between any such two nodes is the difference of their semantic mass if they are on the same axis, or the addition of the two distances calculated from each node to a common node where two axes meet if the two original nodes are on different axes. It is easy to prove that the proposed distance measure also satisfies the properties of a metric.

3. Evaluation

3.1. Task Description

It would be reasonable to evaluate the performance of machine measurements of semantic similarity between concepts by comparing them with human ratings on the same setting. The simplest way to implement this is to set up an experiment to rate the similarity of a set of word pairs, and examine the correlation between human judgement and machine calculations. To make our experimental results comparable with other previous experiments, we decided to use the same sample of 30 noun pairs that were selected in an experiment when only human subjects were involved (Miller and Charles 1991), and in another more recent experiment when some computational models were constructed and compared as well (Resnik 1995). In

fact, in the Resnik (1995) experiment, he replicated the human judgements on the same set of word pairs that Miller and Charles did. When the correlation between his replication and the one done by Miller and Charles (1991) was calculated, a baseline from human ratings was obtained for evaluation, which represents an upper bound that one could expect from a machine computation on the same task. In our experiment, we compare the proposed model with the node-based Information Content model developed by Resnik (1995) and the basic edge-based edge counting model, in the context of how well these perform against human ratings (i.e. the upper bound).

For consistency in comparison, we will use semantic similarity measures rather than the semantic distance measures. Hence our proposed distance measure needs to be converted to a similarity measure. Like the edge counting measure in equation 10, the conversion can be made by subtracting the total edge weights from the maximum possible total edge weights. Note that this conversion does not affect the result of the evaluation, since a linear transformation of each datum will not change the magnitude of the resulting correlation coefficient, although its sign may change from positive to negative.

3.2. Implementation

The noun portion of the latest version (1.5) of WordNet was selected as the taxonomy to compute the similarity between concepts. It contains about 60,000 nodes (synsets). The frequencies of concepts were estimated using noun frequencies from a universal semantic concordance SemCor (Miller et al. 1993), a semantically tagged text consisting of 100 passages from the Brown Corpus. Since the tagging scheme was based on the WordNet word sense definition, this enables us to obtain a precise frequency distribution for each node (synset) in the taxonomy. Therefore it avoids potentially spurious results in occasions when only word (not word sense) frequencies are used (Resnik 1995). The downside of using the SemCor data is the relatively small size of the corpus due to the need to manually tag the sense for each word in the corpus. Slightly over 25% of the WordNet noun senses actually appeared in the corpus. Nevertheless, this is the only publicly available sense tagged corpus. The MLE method would seem unsuitable for probability estimation from the SemCor corpus. To circumvent the problem of data sparseness, we use the Good-Turing estimation with linear interpolation.

3.3. Results

Table 1 lists the complete results of each similarity rating measure for each word pair. The data on human ratings are from the publication of previous results (Miller and Charles 1991, Resnik 1995). Notice that two values in Resnik's replication are not available, as he dropped two noun pairs in his experiment since the word *woodland* was not yet in the WordNet taxonomy at that time. The correlation values between the similarity ratings and the mean ratings reported by Millers and Charles are listed in Table 2. The optimal parameter settings for the proposed similarity approach are: $\alpha=0.5$, $\beta=0.3$. Table 3 lists the results of the correlation values for the proposed approach given a combination of a range of parameter settings.

Word Pair		M&C means	Replication means	Sim_{edge}	Sim_{node}	Sim_{dist}
car	automobile	3.92	3.9	30	10.358	30
gem	jewel	3.84	3.5	30	17.034	30
journey	voyage	3.84	3.5	29	10.374	27.497
boy	lad	3.76	3.5	29	9.494	25.839
coast	shore	3.7	3.5	29	12.223	28.702
asylum	madhouse	3.61	3.6	29	15.492	28.138
magician	wizard	3.5	3.5	30	14.186	30
midday	noon	3.42	3.6	30	13.558	30
furnace	stove	3.11	2.6	23	3.527	17.792
food	fruit	3.08	2.1	24	2.795	23.775
bird	cock	3.05	2.2	29	9.122	26.303
bird	crane	2.97	2.1	27	9.122	24.452
tool	implement	2.95	3.4	29	8.84	29.311
brother	monk	2.82	2.4	25	2.781	19.969
crane	implement	1.68	0.3	26	4.911	19.579
lad	brother	1.66	1.2	26	2.781	20.326
journey	car	1.16	0.7	0	0	17.649
monk	oracle	1.1	0.8	23	2.781	18.611
cemetery	woodland	0.95	NA	0	0	10.672
food	rooster	0.89	1.1	18	1.03	17.657
coast	hill	0.87	0.7	26	8.917	25.461
forest	graveyard	0.84	0.6	0	0	14.52
shore	woodland	0.63	NA	25	2.795	16.836
monk	slave	0.55	0.7	26	2.781	20.887
coast	forest	0.42	0.6	24	2.795	15.538
lad	wizard	0.42	0.7	26	2.781	20.717
chord	smile	0.13	0.1	20	4.452	17.535
glass	magician	0.11	0.1	22	1.03	17.098
noon	string	0.08	0	0	0	12.987
rooster	voyage	0.08	0	0	0	12.506

Table 1. Word Pair Semantic Similarity Measurement

Similarity Method	**Correlation (r)**
Human Judgement (replication)	0.8848
Node Based (Information Content)	0.7941
Edge Based (Edge Counting)	0.6004
Combined Distance Model	0.8282

Table 2. Summary of Experimental Results (30 noun pairs)

3.4. Discussion

The results of the experiment confirm that the information content approach proposed by Resnik (1995) provides a significant improvement over the traditional edge counting method. It also shows that our proposed combined approach outperforms the information content approach. One should recognize that even a small percentage improvement over the existing approaches is of significance since we are nearing the observed upper bound.

The results from Table 3 conform to our projection that the density factor and the depth factor in the hierarchy do affect (although not significantly) the semantic distance metric. A proper selection of these two factors will enhance the distance estimation. Setting the density factor parameter at $\beta=0.3$ seems optimal as most of the resultant values outperform others under a range of depth factor settings. The optimal depth scaling factor α ranges from 0 to 0.5, which indicates it is less influential than the density factor. This would support the Richardson and Smeaton (1995) argument about the difficulty of the adjustment of the depth scaling factor. Another explanation would be that this factor is already absorbed in the proposed link strength consideration. Overall, there is a small performance improvement (2.1%) over the result when only the link strength factor is considered. Since the results are not very sensitive to the variation in parameter settings, we can conclude that they are not the major determinants of the overall edge weight.

Depth Factor	**Density Factor (β)**			
(α)	β=1.0	β=0.5	β=0.3	β=0.2
α=2	0.79844	0.81104	0.81153	0.80658
α=1	0.80503	0.82255	0.82625	0.82266
α=0.5	0.80874	0.82397	0.82817	0.82509
α=0	0.81127	0.82284	0.82737	0.82411
α=-1	0.81435	0.81598	0.81818	0.81349
α=-2	0.81315	0.80228	0.80118	0.79492

Table 3. Correlation coefficient values of various parameter settings for the proposed approach

Further examinations of the individual results in Table 1 may provide a deeper understanding of the model's performance. The ratings in the table are sorted in descending order based on Miller and Charles (1991) findings. This trend can be observed more or less consistently in four other ratings. However, there are some abnormalities that exist in the results. For example, the pair *'furnace-stove'* was given high similarity values in human ratings, whereas a very low rating (second to the lowest) was found in the proposed distance measure. A further look at their classification in the WordNet hierarchy seems to provide an explanation. Figure 2 depicts a portion of WordNet hierarchy that includes all the senses of these two words. We can observe that *furnace* and *stove* are classified under very distinct substructures. Their closest super-ordinate class is *artifact*, which is a very high level abstraction. It would be more reasonable if the substructure containing *furnace* were placed under the class of *device* or *appliance*. If so the distance between *furnace* and *stove* would have been shorter and closer to humans' judgements. This observation re-enforces our earlier thought that the structure of a taxonomy may generate a bias towards a certain distance calculation due to the nature of its classification scheme.

Table 4 shows calculations of the correlation coefficients based on removing the *'furnace-stove'* pair due to a questionable classification of the concept *furnace* in the taxonomy. The result shows an immediate improvement of all the computational models. In particular, our proposed model indicates a large marginal lead.

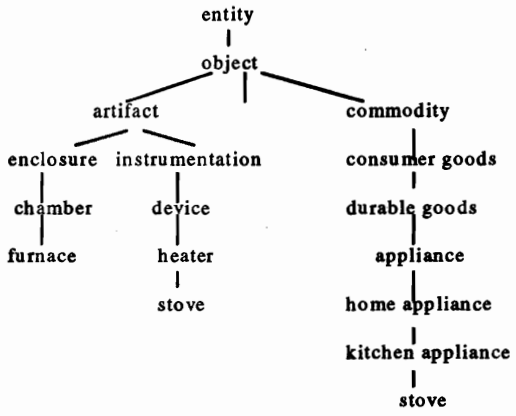


Figure 2. A fragment of WordNet taxonomy

Similarity Method	Correlation (r)
Node Based (Information Content)	0.8191
Edge Based (Edge Counting)	0.6042
Combined Distance Model	0.8654

Table 4. Summary of Experimental Results
(29 noun pairs, removing the *'furnace - stove'* pair)

4. Related Work

Closely related works to this study are those that were aligned with the thread of our discussion. In the line of the edge-based approach, Rada et al. (1989) and Lee et al. (1993) derived semantic distance formulas using the edge counting principle, which were then used to support higher level result ranking in document retrieval. Sussna (1993) defined a similarity measure that takes into account taxonomy structure information. Resnik's (1995) information content measure is a typical representative of the node-based approach. Most recently, Richardson and Smeaton (1995) and Smeaton and Quigley (1996) worked on a combined approach that is very similar to ours.

One of the many applications of semantic similarity models is for word sense disambiguation (WSD). Agirre and Rigau (1995) proposed an interesting conceptual density concept for WSD. Given the WordNet as the structured hierarchical network, the conceptual density for a sense of a word is proportional to the number of contextual words that appear on a sub-hierarchy of the WordNet where that particular sense exists. The correct sense can be identified as the one that has the highest density value.

Using an online dictionary, Niwa and Nitta (1994) built a reference network of words where a word as a node in the network is connected to other words that are its definitional words. The network is used to measure the conceptual distance between words. A word vector is defined as the list of distances from a word to a certain set of selected words. These selected words are not necessarily its definitional words, but rather certain types of representational words called *origins*. Word similarity can then be computed by means of their distance vectors. They compared this proposed dictionary-based distance vector method with a corpus-based co-occurrence vector method for WSD and found the latter has a higher precision performance. However, in a test of leaning positive or negative meanings from example

words, the former gave remarkable higher precision than the latter. Kozima and Furugori (1993) also proposed a word similarity measure by spreading activation on a semantic net composed by the online dictionary LDOCE.

In the area of IR using NLP, approaches have be pursued to take advantage of the statistical term association results (Strzalkowski and Vauthey 1992, Grefenstette 1992). Typically, the text is first parsed to generated syntactic constructs. Then the *head-modifier* pairs are identified for various syntactical structure. Finally, a specific term association algorithm (similar to the mutual information principle) is applied to the comparison process on a single term/concept basis. Although only modest improvement has been shown, the significance of this approach is that it does not require any domain-specific knowledge or the sophisticated NLP techniques. In essence, our proposed combination model is similar to this approach, except that we also resort to extra knowledge sources—machine readable lexical taxonomies.

5. Conclusion

In this paper, we have presented a new approach for measuring semantic similarity between words and concepts. It combines the lexical taxonomy structure with corpus statistical information so that the semantic distance between nodes in the semantic space constructed by the taxonomy can be better quantified with the computational evidence derived from distributional analysis of corpus data. Specifically, the proposed measure is a combined approach that inherits the edge-based approach of the edge counting scheme, which is enhanced by the node-based approach of information content calculation. When tested on a common data set of word pair similarity ratings, the proposed approach outperforms other computational models. It gives the highest correlation value (r=0.828), with a benchmark resulting from human similarity judgements, whereas an upper bound (r=0.885) is observed when human subjects are replicating the same task.

One obvious application of this approach is for word sense disambiguation. In fact, this is part of the ongoing work. Further applications would be in the field of information retrieval. With the lesson learned from Richardson and Smeaton (1995), when they applied their similarity measure to free text document retrieval, it seems that the IR task would benefit most from the semantic similarity measures when both document and query are relatively short in length (Smeaton and Quigley 1996).

References

Agirre, E. and G. Rigau, 1995, "A proposal for Word Sense Disambiguation Using Conceptual Distance", *Proceedings of the First International Conference on Recent Advanced in NLP,* Bulgaria.

Church, K.W. and P. Hanks, 1989, "Word Association Norms, Mutual Information, and Lexicography", *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, ACL27'89, 76-83.

Grefenstette, G., 1992, "Use of Syntactic Context to Produce Term Association Lists for Text Retrieval", *Proceedings of the 15th Annual International Conference on Research and Development in Information Retrieval*, SIGIR'92.

Hindle, D., 1990, "Noun Classification from Predicate-Argument Structures", *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, ACL28'90, 268-275.

Kozima, H. and T. Furugori, 1993, "Similarity Between Words Computed by Spreading Activations on an English Dictionary", *Proceedings of the 5th Conference of the European Chapter of the Association for Computational Linguistics*, EACL-93, 232-239.

Lee, J.H., M.H. Kim, and Y.J. Lee, 1993, "Information Retrieval Based on Conceptual Distance in IS-A Hierarchies", *Journal of Documentation*, Vol. 49, No. 2, 188-207.

Miller, G., 1990, "Nouns in WordNet: A Lexical Inheritance System", *International Journal of Lexicography*, Vol. 3, No. 4, 245-264.

Miller, G., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, 1990, "Introduction to WordNet: An Online Lexical Database", *International Journal of Lexicography*, Vol. 3, No. 4, 235-244.

Miller, G. and W.G. Charles, 1991, "Contextual Correlates of Semantic Similarity", *Language and Cognitive Processes*, Vol. 6, No. 1, 1-28.

Miller, G., C. Leacock, R. Tengi, and R.T. Bunker, 1993, "A Semantic Concordance", *Proceedings of ARPA Workshop on Human Language Technology*, 303-308, March 1993.

Morris, J. and G. Hirst, 1991, "Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of Text", *Computational Linguistics*, Vol. 17, 21-48.

Niwa, Y. and Y. Nitta. 1994, "Co-occurrence Vectors from Corpora vs. Distance Vectors from Dictionaries", *Proceedings of the 17th International Conference on computational Linguistics*, COLING'94, 304-309.

Rada, R., H. Mili, E. Bicknell, and M. Bletner, 1989, "Development and Application of a Metric on Semantic Nets", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 1, 17-30.

Resnik, P., 1992, "WordNet and Distributional Analysis: A Class-based Approach to Lexical Discovery", *Proceedings of the AAAI Symposium on Probabilistic Approaches to Natural Language*, San Joe, CA.

Resnik, P., 1995, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy", *Proceedings of the 14th International Joint Conference on Artificial Intelligence,* Vol. 1, 448-453, Montreal, August 1995.

Richardson, R. and A.F. Smeaton, 1995, *"Using WordNet in a Knowledge-Based Approach to Information Retrieval"*, Working Paper, CA-0395, School of Computer Applications, Dublin City University, Ireland.

Smeaton, A.F. and I. Quigley, 1996, *"Experiments on Using Semantic Distance Between Words in Image Caption Retrieval"*, Working Paper, CA-0196, School of Computer Applications, Dublin City University, Ireland.

Strzalkowski, T. and B. Vauthey, 1992, "Information Retrieval Using Robust Natural Language Processing", *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, ACL'92, 104-111.

Sussna, M., 1993, "Word Sense Disambiguation for Free-text Indexing Using a Massive Semantic Network", *Proceedings of the Second International Conference on Information and Knowledge Management*, CIKM'93, 67-74.

Towards a Representation of Verbal Semantics - An Approach Based on Near Synonyms

Mei-chih Tsai*, Chu-Ren Huang*, Keh-jiann Chen*, Kathleen Ahrens**
*Academia Sinica
**National Taiwan University

Abstract

In this paper we propose using the distributional differences in the syntactic patterns of near-synonyms to deduce the relevant components of verb meaning. Our method involves determining the distributional differences in syntactic patterns, deducing the semantic features from the syntactic phenomena, and testing the semantic features in new syntactic frames. We determine the distributional differences in syntactic patterns through the following five ways: First, we search for all instances of the verb in the corpus. Second, we classify each of these instances into its type of syntactic function. Third, we classify each of these instances into its argument structure type. Fourth, we determine the aspectual type that is associated with each verb. Lastly, we determine each verb's sentential type. Once the distributional differences have been determined, then the relevant semantic features are postulated. Our goal is to tease out the lexical semantic features as the explanation, and as the motivation of the syntactic contrasts.

1. Introduction

Radical Lexicalism maintains that all grammatical behaviors are manifestations of lexical features (Karttunen 1986). Since most lexical attributes are semantic and/or conceptual in nature, taking this lexicon-driven approach to language means that many syntactic properties can be predicted from lexical semantic attributes (Jackendoff 1976, Levin 1985, Dowty 1991, Pustejovsky 1993). In terms of Natural Language Processing (NLP), surface syntactic structures can be systematically predicted from their lexical semantic representation. From this perspective, the automatic acquisition of lexical knowledge for NLP may be possible, since the relation between syntactic patterns and lexical semantics is predictable to some extent. Dorr & Jones (1996), for example,

demonstrate that semantic information can be derived from syntactic cues when the syntactic cues are first divided into distinct groupings that correlate with different word senses.

However, as Levin (1993) points out, there are still many questions to be explored:

> 'If the hypothesis that syntactic properties are semantically determined is taken seriously, then the task is to determine, first, to what extent the meaning of a verb determines its syntactic behavior, and second, to the extent that syntactic behavior is predictable, what components of verb meaning figure in the relevant generalizations. The identification of the relevant components of meaning is essential if this approach is to be successful.' (Levin 1993:14)

Our paper will focus on the last point above. We propose using the distributional differences in the syntactic patterns of near synonyms to deduce the relevant components of verbal semantics. In particular, we want to identify the semantic features that differentiate verbal semantic behaviors. Our strong hypothesis is that all lexical semantic features can be identified this way. In contrast, salient semantic features deduced from a shared verb class may or may not be predictive of verbal features because they may simply be descriptions of the meaning. Our method is as follows:

1) Determine distributional differences in syntactic patterns
2) Deduce the semantic features from the syntactic phenomena
3) Test the semantic features in new syntactic frames

How will we determine the distributional differences in syntactic patterns? Our corpus-based approach calls for us to search, sort, and classify all relevant data according to the four following criteria: First, we will classify each of these instances according to the syntactic functions of the verbs themselves (i.e. predicate, complement, adverbial, determinal, nominal). Second, we can classify the corpus data in terms of argument type that the verbs take (i.e. NP subject, VP subject, sentential subject, NP

object, NP double-object, sentential object). Third, we determine the aspectual types each verb is associated with (i.e. aspectual markers, aspectual adverbs, resultative complements). Lastly, we examine the sentential modes that each verb occurs in (i.e. passive, imperative, evaluative, declarative, interrogative).

This process is time-consuming. However, because we are dealing with near-synonyms, we expect there to be many shared syntactic behaviors that can be ignored for the purpose of this study. This will facilitate the identification of (sometimes unexpected) grammatical contrasts that instantiates deeper lexical semantic contrasts of the near-synonym pairs. The crucial difference will be found in the small number of instances where they are in complementary distribution in terms of one of the above four types of syntactic information. In what follows we will present our 3 - step methodology (i.e. determine syntactic difference, deduce semantic feature, test for reliability of semantic feature) for each of the 4 different types of syntactic information (i.e. syntactic functions (Section 2), argument structure (Section 3), aspectual type (Section 4), sentential type (Section 5)). In the concluding section (Section 6), we discuss the advantages of this method as compared to an account that is based on differentiating semantic classes of verbs (Levin 1993).

2. Syntactic functions

We look at what type of syntactic functions a verb can occur with, including predicate, adverbial, complement, nominalization, etc.

Distributional differences

The distributional contrasts in terms of the syntactic functions between the two state verbs LEI 'be tired' and PIJUAN 'be tired' are that LEI functions as a (resultative) complement in 6% of the cases, but never occurs in a nominal phrase, while PIJUAN serves as a noun in 9% of the instances, but never occurs in a (resultative) complement position. The data from the Academia Sinica Balanced Corpus[1] (abb. Sinica Corpus) is

[1] The Academia Sinica Balanced Corpus is the largest balanced corpus of both written and spoken contemporary Mandarin, developed by CKIP group in Academia Sinica, Taiwan, containing 3.5 million words.

given in Table 1 and the relevant examples are given in (1) and (2). (The numbers next to the verbs in the table indicate the number of instances of occurrence in the entire Sinica Corpus.)

Table 1. Differences in syntactic functions: LEI vs. PIJUAN

Functions	Complement	Nominalization
LEI 174	11 (6%)	--
PIJUAN 33	--	3 (9%)

(1) Resultative complement

(1a) ta zou de hen lei[2]
he walk DE very be-tired
'He walked so much that he was tired.'

(1b) # ta zou de hen pijuan
he walk DE very be-tired

(2) Nominalized object

(2a) shuimian shi zhi pijuan zuihaode fangfa
sleep be treat be-tired best method
'Sleeping in the best method to treat the tiredness.'

(2b) # shuimian shi zhi lei zuihaode fangfa
sleep be treat be-tired best method

Semantic feature

One semantic feature that would distinguish the meaning of these two verbs is [+/-effect]. In other words, though both are states that predicate of people, LEI has the additional meaning that is an effect state of an (unspecified) event, while PIJUAN does not specify this. It is obvious that an effect state occurs as a resultative complement, and represents the effect of another predicate. On the other hand, there seems to be a tendency against nominalized complex verbs in Chinese (e.g. all verb-resultative

[2] The abbreviations used in the glosses are the following: ASP 'aspect marker', BEI 'marker of agent', CL 'classifier', DE 'complement marker', PAR 'sentential-final particle',.

compounds cannot be nominalized). Thus, an effect state has the semantic implicature of a complex event and cannot be nominalized.

Prediction/Verification

After looking at near synonyms to determine the semantic feature that differentiates them, we need to test our hypothesis. The following two examples demonstrate that it is much easier for LEI than for PIJUAN to occur with the sentential-final particle (PAR) *le*. In fact, the statistics shown in Table 2 indicate the relatively high percentage of LEI co-occurring with *le* when compared with the zero utterance of PIJUAN.

(3) Sentential-final particle

(3a) tamen lei le jiu lai ci he pijiu
they be-tired PAR then come here drink beer
'When they are tired, they come here to drink some beer.'

(3b) # tamen pijuan le jiu lai ci he pijiu
they be-tired PAR then come here drink beer

Table 2. Differences in collocations: LEI vs. PIJUAN

Collocation	*le*
LEI 174	42 (24%)
PIJUAN 33	--

As the sentential-final particle primarily signals a change of state (cf. Li & Thompson 1981), the collocation to such an element reveals that the state expressed by LEI is changed from an earlier state. In other words, it is an effective state, i.e. [+ effect]. PIJUAN, on the other hand, is [- effect].

3. Argument selection

The distributional differences for argument selection involve determining whether the verb occurs with an NP subject, VP subject, sentential subject, NP object, double NP object, sentential object, etc.

Distributional differences

In the case of GAOXING and KUAILE 'be happy', GAOXING can take a sentential object in more than 7% of the cases, while KUAILE cannot, as shown in Table 3 and example (4).

Table 3. Differences in argument selection: GAOXING vs. KUAILE

Collocation	Sentential Object
GAOXING 280	20 (7.1%)
KUAILE 365	--

(4) Sentential Object

(4a) tamen hen gaoxing Zhangsan mei zou
they very be-happy John not go-away
'They were glad that John did not go away.'

(4b) # tamen hen kuaile Zhangsan mei zou
they very be-happy John not go-away

Semantic feature

The semantic feature that can be deduced from this distributional difference is one of effect, where GAOXING is an effect state relevant to the cause expressed in the sentential object. It is obvious that an effect state represents the effect brought out by a cause event.

Prediction/Verification

We observe from the data that only GAOXING can be associated with the sentential-final particle *le* in 0.7 % of the instances, as demonstrated below.

(5) Sentential-final particle

(5a)	keren	gaoxing	le	jiu	gei	xiaofei
	customer	be-happy	PAR	then	give	tip
	'When customers are pleased, they give tips.'					
(5b)	# keren	kuaile	le	jiu	gei	xiaofei
	customer	be-happy	PAR	then	give	tip

Table 4. Differences in collocations: GAOXING vs. KUAILE

Collocation	*le*
GAOXING 280	2 (0.7%)
KUAILE 365	--

The contrast between (5a) and (5b) is correctly predicted, because it is possible for GAOXING to represent a changed state triggered off by some cause, while it is not possible for KUAILE.

Thus it is justified to say that GAOXING is an effect state, i.e. [+ effect], whereas KUAILE is [- effect].

4. Aspectual types

The distributional difference for aspectual types involve looking at the aspect markers, aspectual adverbs and resultative complements the verbs co-occur with.

Distributional differences

In the case of QUAN and SHUIFU 'persuade', only QUAN occurs with the durative aspect marker *-zhe*[3] in 1.8% of the cases, SHUIFU never does.

Table 5. Differences in collocations: GUAN vs. SHUIFU

Collocation	*-zhe*
GUAN 112	2 (1.8%)
SHUIFU 50	--

[3] *-Zhe* is also called imperfective aspect marker (Ma 1985, Smith 1985, 1991).

(6) Durative aspect marker

(6a) ta yimian zou, yimian quan-zhe Zhangsan
he one-side walk one-side persuade ASP John
'He persuaded John as he walked.'

(6b) # ta yimian zou, yimian shuifu-zhe Zhangsan
he one-side walk one-side persuade ASP John

Semantic Feature

As the marker *-zhe* indicates that an event is on-going (cf. Li & Thompson 1981), the fact that QUAN can take such a marker and SHUIFU never can suggests that there are aspectual differences between these two verbs. On the one hand, QUAN denotes an extensible, atelic event. On the other hand, SHUIFU denotes a bounded, telic event. The semantic feature that would distinguish the meaning of these two verbs is [+/- telic].

Prediction/Verification

If our hypothesis is correct, we expect that only QUAN is compatible with adverbs indicating the durative aspect. Consider the following examples.

(7) Durative aspectual adverb

(7a) ta yizhi quan Zhangsan jiehun
he all-the-time persuade John get-married
'All the time he persuaded John to get married.'

(7b) * ta yizhi shuifu Zhangsan jiehun
he all-the-time persuade John get-married

The adverb *yizhi* 'all the time' in the above examples can only occur with QUAN but not with SHUIFU. This means that only the event denoted by QUAN can be in progress. The difference between these two verbs in telicity is then justified.

A second argument in support of the claim that QUAN differs from SHUIFU in verbal aspect is related to the fact that only QUAN admits, in 3.6% of instances, resultative

complements which indicate completion or termination (cf. Smith 1991). Consider the examples in (8).

Table 6. Differences in collocations: QUAN vs. SHUIFU

Collocation	Resultative Complement
QUAN 112	4 (3.6%)
SHUIFU 50	--

(8) Resultative complement

(8a) ta quan de Zhangsan xin hen fan
he persuade DE John mood very be-bored
'He kept trying to persuade John until John was bored to death.'

(8b) # ta shuifu de Zhangsan xin hen fan
he persuade DE John mood very be-bored

It is reasonable that telic verbs like SHUIFU excludes the possibility of taking resultative complements, since we cannot terminate an event which is already terminated. But for atelic verbs like QUAN, it is only natural that they take resultative complements, indicating that events are accomplished.

Thus the feature [+/- telic] can account for the contrastive use of aspectual type between these two items.

5. Sentential types:

In this section, we look at what type of sentences a verb can join, including passive sentence, imperative sentence, wish sentence, evaluative sentence, etc.

Distributional differences

One of the distributional contrasts between QUAN and SHUIFU involves the possibility of forming passive sentence. It seems that SHUIFU occurs more frequently in passive construction (6%) than QUAN does (0.9%). The examples in (9) show that QUAN is not allowed in the passive construction without a resultative complement.

Table 7. Differences in collocations: QUAN vs. SHUIFU

Collocation	Passive Sentences
QUAN 112	1 (0.9%)
SHUIFU 50	3 (6%)

(9) Passive sentence

(9a) # Zhangsan bei ta quan le
John BEI he persuade PAR

(9b) Zhangsan bei ta shuifu le
John BEI he persuade PAR
'John was persuaded by him.'

(9c) Zhangsan bei ta quan-zou le
he BEI he persuade go-away PAR
'John was persuaded to leave by him.'

In case of GAOXING and KUAILE 'be happy', the following distributional contrasts in terms of the sentential types are noticed from the Sinica Corpus: GAOXING never constitutes wish sentences but admits evaluational sentences (1.8%), while KUAILE occurs in wish sentences (2.2%) but never appears in evaluational sentences.

Table 8. Differences in collocations: GAOXING vs. KUAILE

Collocations	Wish Sentences	Evaluational Sentences
GAOXING 280	--	5 (1.8%)
KUAILE 365	8 (2.2%)	--

(10) Wish sentence

(10a) zhu ni kuaile!

wish you be-happy

'I wish you be happy.'

(10b) #zhu ni gaoxing!

wish you be-happy

(11) Evaluational sentences

(11a) zhei-jian shi zhide gaoxing.

this CL thing be-worth be-happy

'This thing is worth enjoying.'

(11b) #zhei-jian shi zhide kuaile

this CL thing be-worth be-happy

Semantic Feature

The semantic feature that would distinguish the meaning of QUAN and SHUIFU is [+/-effect]. Though both are events, SHUIFU has an additional meaning of effect which corresponds to the affectedness property of passive sentences, while QUAN does not have.

As for GAOXING and KUAILE, the distinctive feature of their meaning is [+/-control]. Though both are states, GAOXING denotes the meaning of control which accepts the calculated reaction in evaluational sentences and refuses the impredictive nature of wish sentences, while QUAN does not.

Prediction/Verification

The possibility of taking a resultative complement constitutes a good argument for the claim that the meaning of QUAN and SHUIFU can be distinguished by the feature of effect. We have seen in (8) above that this construction is possible for QUAN but not for SHUIFU. How can we account for the fact that QUAN cannot occur in passive sentences alone without a resultative complement behind? The answer is that resultative complements not only indicate the accomplishment of the main event, but also express the affected state of the participant. Thus the use of resultative complements can

contribute to QUAN additional properties like completion and affectedness, which are inherent to SHUIFU.

Now let us turn to the semantic feature [+/- control]. To support the claim that GAOXING can be controlled and KUAILE cannot, consider the use of imperative sentence illustrated below.

Table 9. Differences in collocations: GAOXING vs. KUAILE

Collocation	Imperative Sentences
GAOXING 280	3 (1.1%)
KUAILE 365	--

(12) Imperative sentence

(12a) bie gaoxing!
don't be-happy
'Don't be happy!'

(12b) #bie kuaile!
don't be-happy

The data show that GAOXING can form imperative sentences in 1.1% of the instances, while KUAILE never can. This means that the hearer can only change the state of GAOXING, but not the state of KUAILE. In other words, only the state of GAOXING is controllable.

6. Conclusion

The notion that the syntactic behavior of verbs is semantically determined has been examined extensively, especially for English verbs (please see Levin 1993 for relevant references). The technique that has been used quite productively is one that determines the distinctive behavior of verb classes. Levin summarizes this method:

> The assumption that the syntactic behavior of verbs is semantically determined gives rise to a powerful technique for investigating verb meaning that can be exploited in the development of a theory of lexical knowledge. If the distinctive behavior of verb classes with

> respect to diathesis alternations arises from their meaning, any class of verbs whose members pattern together with respect to diathesis alternation should be a semantically coherent class: its members should share at least some aspect of meaning. Once such a class is identified, its members can be examined to isolate the meaning components they have in common. Thus diathesis alternations can be used to provide a probe into the elements entering into the lexical representation of word meaning. (Levin 1993:14)

However, this technique is not easily implemented in Mandarin, because extensive study of diathesis alternations has not been done in Mandarin. Perhaps one reason is because Mandarin allows both subject and object omission, which means that it is very difficult to get a handle on what is a relevant 'alternation.' The work that has been done on semantic interpretations of syntactic structures (and the verbs that may occur in these structures) in Mandarin, such as in the case of pre-posed objects (such as BA and BEI), while interesting, is inconclusive because the wide variety of contexts and possible meanings defies a unified explanation. (Cf. Thompson 1973, Mei 1978, Bennett 1981, Ren 1991, Sun 1995, etc)

Moreover, the diathesis alternation technique does not allow for a very fine grained analysis of semantic features, because verbs may belong to more than one (seemingly unrelated) alternation class[4], and because different verb classes may share the same alternation[5]. Thus, it is difficult to extract the common semantic feature that predict the difference between the classes. When we look at near-synonyms, on the other hand, we are able to set up a controlled study of lexical semantic contrasts and their grammatical effects. We hope that this fine-grained approach will aid us in identifying the semantic features or attributes that dictate the syntactic differences of verbs.

---

[4] For example, according to Levin (1993), *hit* belongs to verbs of throwing, verbs of contact by impact as well as verbs of existence, whereas *cut* belongs to verbs of cutting, verbs of separating and disassembling, verbs of creation and transformation, verbs of psychological state, verbs of bodily state and damage to the body, verbs of grooming and bodily care and also meander verbs.

[5] For example, *hit* and *cut* share the conative alternation.

References

Bennett, P., "The Evolution of Passive and Disposal Sentences," Journal of Chinese Linguistics 9, 1981, pp. 61-89.

Dorr, B. J., and D. Jones, "Role of Word Sense Disambiguation in Lexical Acquisition: Predicting Semantics from Syntactic Cues," in Proceedings of 16th International Conference on Computational Linguistics (COLING 96), 1996, pp. 322-327.

Dowty, D. R., "Thematic Proto-Roles and Argument Selection," Language 67, 1991, pp. 547-619.

Jackendoff, R. S., "Towards an Explanatory Semantic Representation," Linguistic Inquiry 7, 1976, pp. 89-150.

Karttunen, L., Radical Lexicalism, CSLI-86-68.

Levin, B. (Ed.), Lexical Semantics in Review, Lexicon Project Working Papers 1, Center for Cognitive Science, MIT, 1985.

Levin, B., English Verb Classes and Alternations: A Preliminary Investigation, University of Chicago Press, 1993.

Liu, M., "Lexical Meaning and Discourse Patterning - The three Mandarin cases of 'build'," paper presented in the 3rd Conference on Conceptual Structure, Discourse, and Language, Boulder CO., 1997.

Ma, J.-H., A Study of the Mandarin Chinese Verb Suffix *ZHE*, Crane Publishing. 1985.

Mei, K., "*Ba* Sentences in Mandarin Chinese," Wen-shi-zhe Xuebao 27, 1978, pp. 145-180. Taiwan University, Taipei.

Pustejovsky, J., The Generative Lexicon, MIT Press, 1993.

Pustejovsky, J., S. Bergler, and P. Anick, "Lexical Semantic Techniques for Corpus Analysis," Computational Linguistics 19.2, 1993, pp. 331-358.

Ren, X., "The Post-Verbal Constituent in Chinese Passive Forms," Journal of Chinese Linguistics 19, 1991, 221-241.

Smith, C. S., "Notes on Aspect in Chinese," Texas Linguistic Forum 26, 1985, pp. 1-36.

Smith, C. S., The Parameter of Aspect, Kluwer Academic Publisher, 1991.

Sun, C., "Transitivity, the *Ba* Construction and Its History," Journal of Chinese Linguistics 23, 1995, pp. 159-195.

Thompson, S. A., "Transitivity and the *Ba* Construction in Mandarin Chinese," Journal of Chinese Linguistics 15.1, 1973, pp. 208-221.

Tsai, M.-C., C.-R. Huang, and K.-J. Chen, "You jinyici bianyi biaozhun kan yuyi jufa zhi hudong. (From near-synonyms to the interaction between syntax and semantics)," in Proceedings of 5th International Symposium on Chinese Languages and Chinese Linguistics (IsCLL 5), 1996, pp. 167-180.

Word Sense Disambiguation Based on The Information Theory

Ho Lee, Dae-Ho Baek, Hae-Chang Rim
Natural Language Processing Lab.,
Department of Computer Science and Engineering,
Korea University,
Anam-Dong, Seoul 136, Republic of South Korea
leeho@nlp.korea.ac.kr daeho@nlp.korea.ac.kr rim@nlp.korea.ac.kr

Abstract

The task of word sense disambiguation is to identify the correct sense of a word in context. In this paper, we define a new notion, classification information, based on the Shannon's information theory. The classification information of a word consists of the pair of the most probable class MPC and the discrimination score DS. In the sense decision of the target word, the MPC of a surrounding word represents the sense of the target word most closely related, and the DS represents the degree of correlation between the MPC and the surrounding word. When a new sentence containing the target polysemous word is given, the sense of the target word is determined to the most plausible sense based on the classification information of all surrounding words in the sentence. Experimental results show that the average accuracy of the proposed method is 84.6% for the Korean data set, and 80.0% for the English data set.

1. Introduction

The task of word sense disambiguation is to identify the correct sense of a word in context. The different meanings of a word are listed as its various senses in a dictionary. The improvement in the accuracy of identifying the correct word sense will result in better machine translation systems, information retrieval systems, etc.(Ng 1996).

There have been many approaches to solve word sense disambiguation problem. In the earlier, (Kelly 1975) and (Weiss 1973) made use of hand-coded knowledge. Therefore, it is nearly impossible to apply those approaches to practical systems because it is quite labor intensive to construct rules manually in those approaches(Gale 1992).

Recently, various knowledge sources have been utilized to resolve word sense ambiguity. One group acquired knowledge from machine readable dictionaries, and the other group acquired knowledge from sense tagged corpora. The first group of

researchers, (Lesk 1986), (Walker 1987), (Luk 1995), and (Ide 1990), use machine readable dictionaries, such as *Oxford's Advanced Learner's Dictionary of Current English*, to resolve word sense ambiguity. They try to develop a program that can read an arbitrary text and tag each word in the text with a pointer to a particular sense number in a particular dictionary. However, those approaches do not seem to work very well because dictionaries simply do not record enough of the relevant information.

The second group, such as (Miller 1994), (Leacock 1993), (Yarowsky 1992), (Bruce 1994), and (Ng 1996), acquired knowledge from a sense tagged corpus in order to solve word sense disambiguation problem. They extracted unordered set of surrounding words, part of speech of target words, morphological forms, or syntactic relations from corpus. In order to employ those extracted information, they used statistical classifiers, neural networks, IR-based techniques, or exemplar-based learning method. The approaches based on a sense-tagged corpus can reduce human intervention, and report relatively high accuracy.

Recently, there are a few approaches to overcome knowledge acquisition bottleneck problem. Yarowsky(1995) proposed an unsupervised training method, and Gale (1992) used a bilingual corpus in order to solve knowledge acquisition bottleneck problem.

In this paper, we propose a method of resolving word sense ambiguity based on minimal information extracted from a sense tagged corpus. For this research, we define the classification information which can be represented by the most probable class(henceforth, *MPC*) and the discrimination score(henceforth, *DS*).

This paper is organized as follows. In the following section, we define the classification information. In the section 3, we apply the classification information to the word sense disambiguation problem, and then we show the experimental results in the section 4. Finally, we discuss the characteristics and problems of our method, and present the possible way of overcoming the problems in future.

2. Classification Informations

In this section, we define the classification information to determine the sense of the target word. The classification information is formalized form of information involved in each surrounding word. The classification information of a surrounding word consists of two fields, the *MPC* and the *DS*. The *MPC* of a surrounding word represents the sense of the target word most closely related, and the *DS* represents the degree of correlation between the *MPC* and the surrounding word.

Shannon(1951) understood information as a liberty of choice. The liberty of choice is granted on a selected message among various messages which can be

produced by information sources. He thinks that the uncertainty grows in proportion to the amount of the increased liberty. Moreover, he measured the uncertainty by the entropy, and the measure becomes the average information value per message. The information value of the i-th message in the entropy equation is $\log_2 p_i$, which is determined by p_i, the occurrence probability of the message. So entropy, H, becomes the average information value of n messages.

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

From the viewpoint of the information theory, each surrounding word can decrease the uncertainty of the given target word. The word, which can decrease much uncertainty, has more discriminating ability. Therefore, assuming that the size of data for each sense is the same, the noise produced by the surrounding word w_k is defined as

$$\begin{aligned} noise_k &= -\sum_{i=1}^{n} p(sense_i|w_k) \log_2 p(sense_i|w_k) \\ &= -\sum_{i=1}^{n} \frac{freq(sense_i, w_k)}{freq(w_k)} \log_2 \frac{freq(sense_i, w_k)}{freq(w_k)} \end{aligned} \tag{2}$$

where n is the number of senses, and p_i, the occurrence frequency of surrounding word w_k, represents $p(sense_i|w_k)$, the conditional probability of $sense_i$ given the surrounding word w_k. In the equation (2), $noise_k$ has the value from 0 to $\log_2 n$ and it has maximum value when all occurrence probabilities of w_k are same. The word whose noise is high has low discriminating ability and provides little assistance for determining the sense of the target polysemous word. Therefore, we can measure the discriminating ability with the reverse function of noise as shown in the equation (3).

$$DS_k = signal_k = \log_2 n - noise_k \tag{3}$$

The MPC can be calculated according to the equation (4).

$$MPC_k = \text{argmax}_i\ p_i = \text{argmax}_i\ p(sense_i|w_k) = \text{argmax}_i\ \frac{freq(sense_i, w_k)}{freq(w_k)} \tag{4}$$

The equations (3) and (4) are based on the hypothesis that the size of data for each sense is same. However, the difference among the size of data may have an effect on the values of the MPC_k and the DS_k. Therefore, the normalization based on the data size is required. The normalized occurrence probability $\hat{p}_i$ is defined as the equation (5).

$$\hat{p}_i = \frac{p_i \frac{N(senses)}{N(sense\ i)}}{\sum_{j=1}^{n} p_j \frac{N(senses)}{N(sense\ j)}} = \frac{p(w_k|sense\ i)}{\sum_{j=1}^{n} p(w_k|sense\ j)} \tag{5}$$

where $N(sense_i)$ represents the data size of i-th sense, and $\overline{N(senses)}$ represents the average of $N(sense_i)$. The equation (6) shows the modified formula of $noise_k$ based on the equation (5).

$$noisy_k = -\sum_{i=1}^{n} \hat{p}_i \log_2 \hat{p}_i = -\sum_{i=1}^{n} \frac{p(w_k|sense\ i)}{\sum_{j=1}^{n} p(w_k|sense\ j)} \log_2 \frac{p(w_k|sense\ i)}{\sum_{j=1}^{n} p(w_k|sense\ j)} \quad (6)$$

In the equation(6), $noise_k$ also has the value from 0 to $\log_2 n$. The normalized DS_k can be calculated by applying the equation (6) to the equation (3). The normalized MPC_k can be acquired by the equation (7).

$$MPC_k = argmax_i\ \hat{p}_i \quad (7)$$

3. Sense Decision Using Classification Information

With the following sentence, we will explain the import of the classification information in the word sense disambiguation.

Several financial institutions, both ***banks*** *and insurance companies, have been sounded out.*

In general, human refers surrounding words in order to determine the sense of the polysemous word 'bank'. However, not all of the surrounding words can provide clues for the sense decision. The surrounding words, 'financial', 'institution', 'insurance', and 'company' provide important clues. On the other hand, 'several', 'have', 'be', 'sound', and 'out' provide less information to the sense decision. The words providing important clues occur frequently in the sentence that the word 'bank' is used as one specific sense, but occur rarely in the sentence that the word 'bank' is used as other senses. Consequently, important clues have high DS value in the classification information.

Because the classification information provides the importance of the surrounding word, we can easily determine the sense of the target word with the summation of DS of all surrounding words. The sense of the target word contained in a sentence $S = \{w_1, w_2, \cdots w_n\}$ can be determined by the equation (8).

$$MPC(S) = argmax_i \sum_{k=1}^{n} DS_k(i) \quad (8)$$

where the discrimination score of w_k over $sense_i$, $DS_k(i)$, is defined as the equation (9).

$$DS_k(i) = \begin{cases} DS_k & \text{if } i = MPC_k \\ 0 & otherwise \end{cases} \quad (9)$$

For example, the table 1 presents the sense decision in a sentence containing

surrounding words	Training phase		Testing phase			
	MPC_k	DS_k	$DS_k(i)$			
			sense 1	sense 2	sense 3	sense 4
w_1	3	0.7324	0	0	0.7324	0
w_2	2	1.3881	0	1.3881	0	0
w_3	2	0.9077	0	0.9077	0	0
w_4	4	0.3140	0	0	0	0.3140
w_5	3	0.2663	0	0	0.2663	0
w_6	1	0.5817	0.5817	0	0	0
w_7	2	0.8203	0	0.8203	0	0
w_8	3	0.4938	0	0	0.4938	0
$\sum_{i=1}^{8} DS_k(i)$			0.5817	3.1161	1.4925	0.3140
sense of the target word			sense 2			

Table 1. An example of sense decision using the classification information

words $w_1 \sim w_8$. The DS_k is assigned to $DS_k(i)$ if i is the MPC of w_k, and 0 otherwise. Therefore the value of $DS_1(3)$ becomes 0.7324 and other values of $DS_1(i)$ becomes 0, because the MPC of w_1 is $sense_3$ and the DS of w_1 is 0.7324. Finally, we determine the sense which has the maximum $\sum_i DS_k(i)$ as the most plausible sense of the target word.

4. Experimental Results

Our word sense disambiguation method is tested with the data from two languages, one is Korean and the other is English. Probably, our method can be applied to any other language because only the occurrence frequencies of surrounding words are required to determine the word sense.

4.1 Korean Word Sense Disambiguation

Words	Senses
배(Pae):NN	the belly(腹), a pear(梨), a boat(船), an embryo(胚)
전자(Jeon-Ja):NN	an electron(電子), the former(前者)
감다(Kam-Ta):VV	close one's eyes, wash, wind
열리다(yeol-Ri-da):VV	open, hold a meeting, spread a space, make way for a person, start up, enlighten, make out what a person say

Table 2. Four Korean polysemous words and their senses

Word	Inside test			Outside test		
	baseline	accuracy	improvement	baseline	accuracy	improvement
배(Pae)	61.4%	92.8%	31.4%	69.6%	78.3%	8.7%
전자(Jeon-Ja)	87.3%	98.0%	10.7%	69.5%	81.0%	11.5%
감다(Kam-Ta)	60.3%	98.4%	38.1%	80.8%	84.9%	4.1%
열리다(yeol-Ri-da)	68.8%	100.0%	31.2%	70.3%	81.6%	11.3%

Table 3. The results of inside and outside test

For the first experiment, we select four target polysemous words, extract concordances of those words from 10 million size raw corpus, and manually tag the sense of the word. In the outside test, we select the 80% of the concordances as a training set and the remaining concordances as a test set. The table 2 contains the target polysemous words and their senses.

The table 3 contains the result of the inside test and the outside test acquired from 100 trials. The baseline method in the table 3 represents the primitive method that always selects the most frequent sense. In the inside test, the accuracy of our method is much higher than the baseline method. From this result, we can say that the classification informations reflect the implicit informations of the training data set very well. However, the average accuracy in the outside test is about 84.6%. We think that one major reason of the low accuracy is the data sparseness. We also think that morphological ambiguity has bad effect on word sense disambiguation since we use the raw corpus for training and testing.

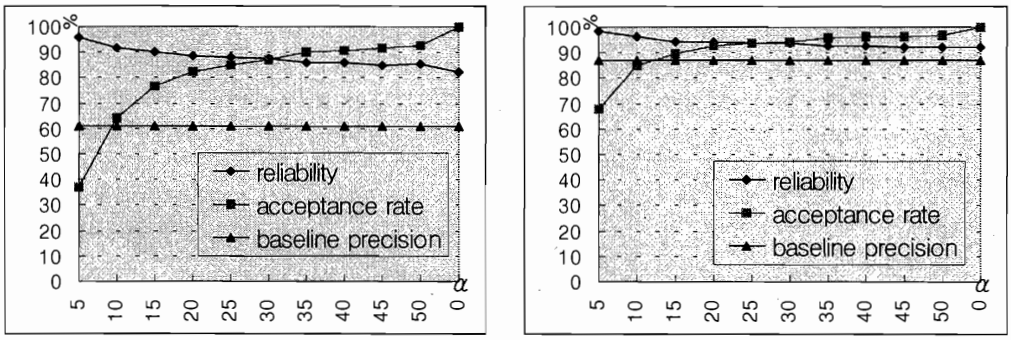
The table 4 shows the average difference between the DS of the correct sense and the maximum DS of incorrect sense per word. The values in the table 4 are calculated by the equation (10)where N is the number of words in the sentence and cs denotes the correct sense.

$$\frac{\left| \sum_{k=1}^{N} DS_k(cs) - \{ argmax_{i, i \neq cs} \sum_{k=1}^{N} DS_k(i) \} \right|}{N} \tag{10}$$

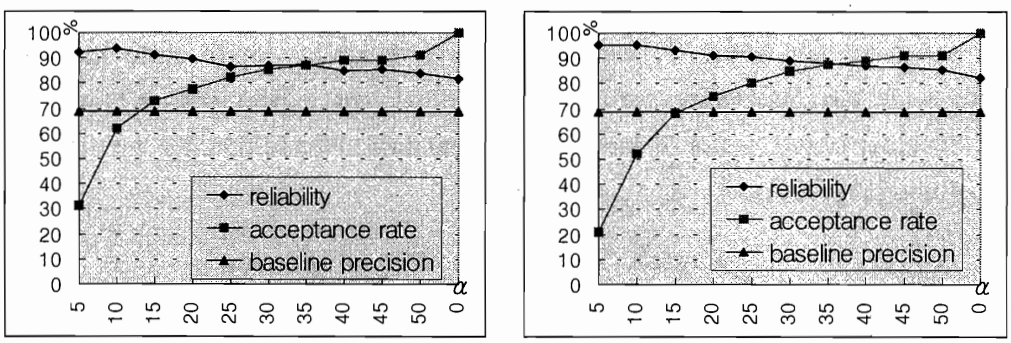
As shown in the table 4, the average differences of DSs are much smaller in the case that incorrect senses are selected. We have made an experiment that admit

Words	Successful case	Failed case
배(Pae)	0.2905	0.1337
전자(Jeon-Ja)	0.1595	0.0936
감다(Kam-Ta)	0.2683	0.1062
열리다(yeol-Ri-da)	0.3017	0.1360

Table 4. The differences between $DS_k(i)$



(a) 배(Pae):NN (b) 전자(Jeon-Ja):NN



(c) 감다(Kam-Ta):VV (d) 열리다(yeol-Ri-da):VV

Figure 1. The reliability and the acceptance rate of reservation strategy

the *empty decision*. The empty decision represents the case when the word sense decision is deferred if the average difference of *DS* per word is less than the threshold calculated by the equation (11).

$$Threshold = \frac{\log_2 n}{\alpha} \tag{11}$$

where α is a arbitrary constant value and n is the number of senses. In the equation (11), we do not use the single constant value as the threshold. The more sense the polysemous word has, the greater value the average difference of *DS* per word has. Therefore, the empty decision rate increases in proportion to the number of senses, if the threshold has the single constant value. In order to acquire the consistent result for all polysemous words, we make the variable threshold in proportion to the maximum of the average difference of *DS* per word. For example, if the value of α is 5, then the empty decision breaks out when the average difference of *DS* per word is less than $\frac{1}{5}$(=20%) of the maximum value.

The figure 1 show the experimental results of the reservation strategy. The reliability means the proportion of the correct decision to the total number of decision. The acceptance rate means the proportion of the decided sentences to whole input sentences.

WSD research	accuracy
baseline	53%
Black(1988)	72%
Zernik(1990)	70%
Yarowsky(1992)	72%
Bruce & Wiebe(1994)	79%
Ng & Lee(1996)	89%
proposed method	80%

Table 5. Comparison with previous works

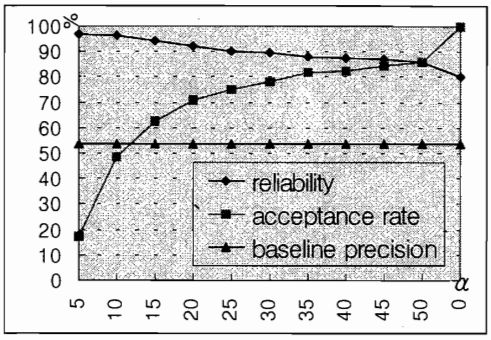


Figure 2. The result of reservation strategy - interest:NN

As shown in the figure 1, we can improve reliability by a little loss of acceptance rate with the reservation strategy. Therefore, we expect that we will get high accuracy if other word sense disambiguation method is additionally employed to our method as a post-process.

4.2 English Word Sense Disambiguation

In the second experiment, we used an English data set which has been commonly used in several previous researches. So far, very few existing works on word sense disambiguation have been tested and evaluated on a common data set. We could acquire only one sense-tagged data set used in (Bruce 1994), which has been made available in the public domain by Bruce and Wiebe. The data set consists of 2369 sentences each containing an occurrence of the noun "interest" (or its plural form "interests") with its correct sense manually tagged(Bruce 1994)(Ng 1996). In order to compare our method with other researches, we applied classification informations to the common data set. The results of previous researches and our approach are shown in table 5.

As shown in the table 5, our proposed method is relatively better than previous works except the Ng's method. Ng's method is better than any other method in terms of the accuracy because he used complex informations such as parts of speech and surface forms of target words, surrounding words, collocations and structural relations. In our approach, however, only surrounding words are used to determine word senses. Therefore, our approach can be easily applied to other languages.

We also apply the reservation strategy to the English data set, and the result is shown in the figure 2. We can also achieve high reliability by a little loss of acceptance rate in the English data set by admitting the empty decision.

5. Conclusions and Future Works

In this paper, we have presented a method of word sense disambiguation by using

classification informations. We have achieved about 96.7% accuracy in the inside test and about 84.6% accuracy in the outside test. Moreover, we could achieve higher accuracy at the cost of few recall rate under the reservation strategy.

We can say that our method has three characteristics. The first characteristic is the ease of modeling. As we use classification informations, it is possible to decompose whole word sense disambiguation model easily into word unit models. The second characteristic is the ease of information acquisition. For classification informations of word sense disambiguation, the minimal information, the occurrence frequencies of surrounding words, is only required. The third characteristic is language independency. However, our method can be applied to any other language because the information used in our method is so simple that it can be extracted by the same procedure regardless of the language. Our method have two problems, the knowledge acquisition bottleneck and the data sparseness problem.

For the future work, we will try to use a word class as a unit of the classification information in order to solve the data sparseness problem and combine our method to the unsupervised training technique. Moreover, we will also study the technique of combining classification informations with other useful informations.

References

Black, Ezra, "An Experiment in Computational Discrimination of English Word Sense," *IBM Journal of Research and Development*, 32(2), pp. 185-194, 1988.

Bruce, R. and Wiebe, J., "Word Sense Disambiguation using Decomposable Models," In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 139-146, 1994.

Gale, W., Church, K. and Yarowsky, D., "A Method for Disambiguating word senses in a large corpus," *Computers and the Humanities*, 26, pp. 415-439, 1992.

Ide, N. M., and Veronis J., "Very Large Neural Networks for Word Sense Disambiguation,", In *Proceedings of the 9th European Conference on Artificial Intelligence, ECAI90*, pp. 366-368, 1990.

Kelly, E. and Stone, P., *Computer Recognition of English Word Senses*, 1975.

Leacock, C., Towell, G., and Voorhees, E., "Corpus-based statistical sense resolution," In *Proceedings of the ARPA Human Language Technology Workshop*, 1993.

Lesk, M., "Automatic Sense Disambiguation: How to tell a Pine Cone from an Ice Cream Cone," In *Proceeding of the 1986 SIGDOC Conference*, 1986.

Luk, K. A., "Statistical sense disambiguation with relatively small corpora using dictionary definitions," In *Proceedings of the 33rd Annual Meetings of the Association for Computational Linguistics*, pp. 181-188, 1995.

Miller, A. G., Chodorow, M., Landes, S., Leacock, C. and Robert G. T., "Using a

semantic concordance for sense identification," In *Proceedings of the ARPA Human Language Technology Workshop*, 1994.

Ng, H. T. and Lee, H. B., "Integrating Multiple Knowledge Sources to Disambiguate Word Sense: An Exemplar-Based Approach," In *Proceedings of the 34th Annual Meetings of the Association for Computational Linguistics*, pp. 40-47, 1996.

Shannon, C. E., Prediction and Entropy in Printed English, In *Bell System Technical Journal*, pp. 50-65, 1951.

Walker, D., "Knowledge Resource Tools for Accessing Large Text Files," In *Machine Translation: Theoretical and Methodological Issues*, 1987.

Weiss, S., "Learning to Disambiguate.", In *Information Storage and Retrieval*, pp. 33-41, 1973.

Yarowsky, D., "Word-sense Disambiguation using statistical models of Roget's categories trained on Large Corpora," In *Proceedings of the Fifteenth International Conference on Computational Linguistics*, pp. 456-460, 1992.

Yarowsky, D., "Unsupervised Word Sense Disambiguation rivaling supervised methods," In *Proceedings of the 33rd Annual Meetings of the Association for Computational Linguistics*, pp. 189-196, 1995.

Zernik, Uri, "Tagging word senses in corpus:the needle in the haystack revisitied," *Technical Report 90CRD198*, GE R&D Center, 1990.

An Agreement Error Correction Method Based on a Multicriteria Approach : An Application to Arabic Language

BELGUITH HADRICH Lamia, BEN HAMADOU Abdelmajid, ALOULOU Chafik
Faculté des Sciences Economiques et de Gestion de Sfax
Laboratoire de recherche LARIS, B.P. 1088, 3018 - Sfax - TUNISIE
Tél. (2164) 278 777, Fax (2164) 279 139

Abstract

Most parsers handling syntactic agreement detect the errors but rarely give enough information on how to correct them. Our interest here is the agreement error correction. Thus, we suggest a multicriteria approach to guide the choice of the best alternative. We propose three main criteria (frequency criterion, morphological criterion and typographic criterion) which we apply to Arabic sentences in order to evaluate the alternatives, and we show the interest of TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) as an aggregation method for the proposed criteria.

Key Words: Agreement error detection, agreement error correction, multicriteria approach, TOPSIS, correction alternatives ranking.

1 Introduction

Many studies have dealt with the problem of agreement errors in written texts. Most of them addressed the detection process rather than the correction one.

The correction process involves the problem of choosing the proper correction among several alternatives. The first parsers left this choice to the user ((Ravin 88), (Coch and Morise 90)) although it can be done automatically and without hesitation in some cases. Therefore, the next parsers opted for the automation of the process of choosing the appropriate correction by using criteria to rank the correction alternatives. Thus the user is guided by the parser in order

to choose the appropriate correction ((Lapalme and Richard 86), (Veronis 91), (Bolioli and al 92)).

The first criterion proposed to classify the possible corrections gives priority to the head of the phrase ((Strube 90), (Genthial and al 90), (Genthial and al 94)) : the idea here is that the writer takes more care to the main words (the governors) than the others (the dependants).

It is clear that this criterion is irrelevant if we deal with competence errors. In French these errors are mainly omission or addition of silent morphological marks (e.g., the mark "s" of the plural) (Veronis 88). In such cases, the governor can't be used for the correction even if the user gives a particular care to the main words.

Moreover, if we consider the case where the agreement marks of the dependants are more frequent than that of the governor, it is unfair to impose the correction according to the governor features only.

These works prove that taking into account one criterion is not appropriate to differing phrases. We think that more one criterion must be considered. The multiplicity of criteria can handle the different causes of errors.

Our study of the Arabic language proves that we can choose three main criteria (the frequency criterion, the typographic criterion and the morphological criterion).

This paper focuses the use of a multicriteria approach to classify the possible corrections in order to choose the best one. This approach can be applied to any language even if in this paper we choose the Arabic language.

We first present a brief overview of the method used to detect the agreement errors in Arabic sentences. Then we propose three main criteria to evaluate the correction alternatives and we present the techniques of scoring them. Finally, we show the interest of using TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) as an aggregation method of these criteria.

2 Agreement error detection method

Most of studies agree with the necessity of parsing the apprehended text in order to detect the agreement errors ((Blache 90), (Blache 91), (Genthial and Courtin 92)). But we think that the accuracy and the robustness of the parser strongly depend on the typology of errors to be

handled. Thus, to detect the past participle agreement errors, a robust parser is required in order to identify the correct syntactic dependencies (Lapalme and Richard 86). Whereas, to handle agreement errors in gender and number, we think that a partial analysis can be sufficient. We proposed in ((Ben Hamadou and Belguith 96a), (Belguith and Ben Hamadou 96b)) a global analysis approach applied to Arabic and termed "Extended Syntagmatic Analysis". This approach aims to group, in the same sets, all the units of the phrase concerned by at least one agreement rule. The resulting sets are termed the "Extended Syntagms".

The proposed approach is based on two main steps which may be summarised as follows :

Step 1: Identification of the initial syntagms

The initial syntagms are mainly identified by the location of the "Function words" (i.e., the particles, the prepositions, etc.). These words are used to identify the syntagm boundaries. We can distinguish three categories of "Function words":

- words which separate two consecutive syntagms and do not belong to any one of them (e.g., prepositions, coordinating conjunctions, etc.)
- words which start a syntagm and belong to it: this is the case of demonstrative pronouns, relative pronouns, etc.
- words which end a syntagm and are referred to previous words which do not belong to this syntagm (i.e., possessive pronouns).

Step 2: Constitution of the "Extended Syntagms"

The constitution of the "Extended Syntagms" is guided by a rule set which aims to extend the initial syntagms by all the units of the phrase (function words or initial syntagms) that have a dependency relationship.

The result is a list of independent syntagms in which we can, separately, apply the process of agreement error detection. The detection process can be reduced to a simple unification process of the morphological features of all the constituents of the extended syntagm.

Example :

Let us consider the sentence : 'إعتنى الممرّضة المتربّصات بالمريض وأعطته الدّواء'

↑3 ↑2 ↑1

(The trainee$_{fem.plu.}$ nurse$_{fem.sing.}$ took care $_{masc.sing.}$ of the patient$_{masc.sing.}$ and gave $_{fem.sing.}$ him$_{masc.sing}$ some medicines)

The location of the function words: "ب" (1) , "و" (2) ," ه" (3) entails the decomposition of the sentence into the following initial syntagms :

SI_1={ المتربّصات (trainee) , الممرّضة (nurse), إعتنى (took care) }

SI_2= { المريض (patient) }

SI_3= { أعطت (gave) }

SI_4= { ه (him) }

SI_5= { الدّواء (medicines) }

The result of the "Extended Syntagmatic Analysis" is given by :

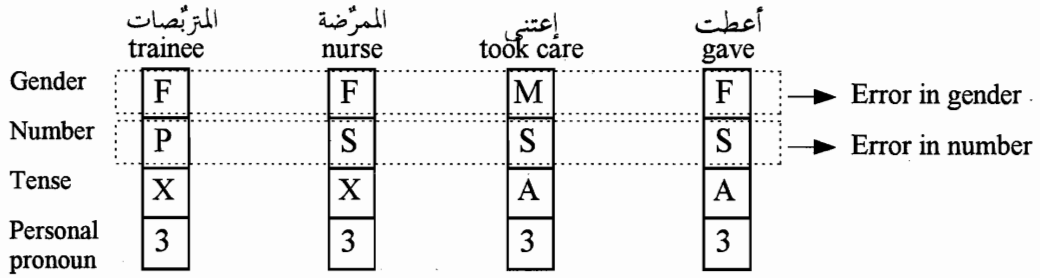
SE_1={ أعطت (gave), المتربّصات (trainee) , الممرّضة (nurse), إعتنى (took care) }

SE_2={ ه (him), المريض (patient) }

SE_3= { الدّواء (medicines) }

The detection process can be done separately in each extended syntagm[1].

Let us consider SE_1. The units of SE_1 and their features can be represented by the following figure[2] :



The unification process of the unit features of SE_1 fails in terms of gender and number.

3 Agreement error correction method

Upon many correction alternatives, the choice of the best alternative may be obvious : this is generally the case of phrases involving few errors since the best solution is the same in terms of all points of view. For instance, the sentence 'الأولاد الذين يلعب في الحديقة تلامذتي' (the

1 SE_3 is a singleton, so it is not concerned by the detection process.

2 Gender : F(Feminine), M(Masculine).
Number : S(Singular), P(Plural), D(Duel)
Where Duel refers to two persons and plural refers to more than two persons.
Tense : P(Present). F(Future), A(Past).
Personal pronoun : 1.2,3.

children$_{masc.pl}$ who plays$_{masc.sing}$ in the garden are$_{masc.pl}$ my students $_{masc.pl}$) has two possible corrections : the first one aims to line up the sentence with the singular, however the second one favors the plural.

To classify these corrections, a first point of view consists of minimizing the number of errors and therefore aims to favor the correction which features are the most frequent. An other point of view may favor the correction that minimizes the number of typographic transformations.

We can remark that the second alternative (plural) is the best one according to the two points of view (three word in the plural and only one in the singular; addition of two letters versus an omission of five letters and a substitution of one letter).

Nevertheless, the best correction is not usually the same according to all points of view. For example, if we consider the sentence 'البنت الصغير النشيط نجحوا في الإمتحان' (The little $_{masc..sing.}$ and dynamic $_{masc..sing.}$ girl $_{fem.sing.}$ succeeded $_{masc.pl.}$ in the exam), the best correction given by the frequency criterion (first point of view) is the masculine-singular. Whereas the typographic criterion (second point of view) favors the correction with the feminine-singular. Consequently, the choice of the best alternative requires a careful analysis and, inevitably, needs a negotiation between the considered criteria. In the following section, we present the basic concepts of a multicriteria approach and we show how it is appropriate to this kind of problems.

3.1 Basic concepts of the multicriteria approach

Our multicriteria decision problem can be defined as follows :

Let $X=\{x_1, \ldots, x_n\}$ a set of correction alternatives and let $F = \{f_1, \ldots, f_q\}$ a set of criteria. The evaluation function of an alternative x_i according to criterion f_j is denoted by :

$$f_j : X \rightarrow IR$$
$$x \rightarrow f_j(x)$$

Each criterion has to be maximized. The problem can be written as follows :

$$\left| \begin{array}{l} \text{"Max" } f(x)= (f_1(x), f_2(x), \ldots, f_q(x)) \\ \\ \text{subject to } x \in X \end{array} \right.$$

We say that x_1 dominates x_2 if $\forall$ i, $f_i(x_1) \geq f_i(x_2)$ $(f(x_1) \neq f(x_2))$.

A correction alternative in X is said efficient if it is not dominated.

In order to determine all efficient correction alternatives, we can use one of the following methods :

1. $P(\alpha) = \text{Max} \sum_j \alpha_j f_j(x) \quad \alpha_j > 0 \ , \quad \sum_{j=1}^{q} \alpha_j = 1$, where α_j is the associated weight of the criterion f_j. The solution of $P(\alpha)$ is an efficient correction alternative.

2. The ideal correction alternative (x^+) is the point in IRq whose cordinates are :

$$(y_1^+, \ldots, y_q^+) \text{ where } y_j^+ = \underset{x}{\text{Max}}\ f_j(x) \quad j = 1,\ldots,q$$

$P(d) = \underset{x}{\text{Min}}\ (d(f(x^+), f(x)))$, where d is a distance (e.g., Euclidean distance).

The solution of P(d) is generally an efficient one.

In our work we will use TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) which is advocated to cardinal criteria and uses a combination of $P(\alpha)$ and P(d) to rank the efficient correction alternatives.

3.2 Main criteria to evaluate the correction alternatives

When the detection of agreement errors involves many correction alternatives, choosing the best one is not usually a simple task since it requires the ranking of the alternatives according to many criteria.

We propose three main criteria to evaluate the correction alternatives in Arabic sentences : the frequency criterion, the morphological criterion and the typographic criterion.

3.2.1 Frequency criterion

The frequency criterion is measured by the occurrence of the alternative features in the sentence. This criterion favors the alternative whose features are more frequent in order to minimize the agreement error number. For example, the sentence :

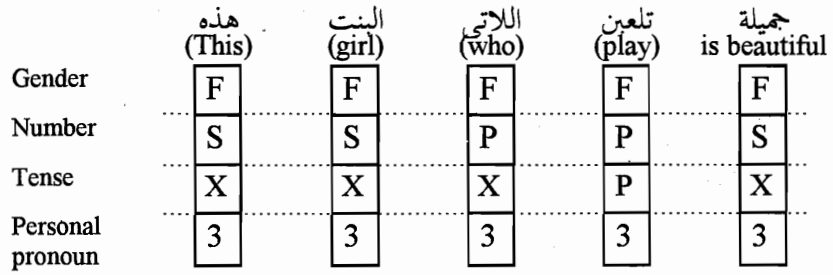
'هـذه البنـت الاتـي تلعـبن في الحديقـة جميلـة جـدا' (This$_{\text{fem. sing.}}$ girl$_{\text{fem.sing.}}$ who$_{\text{fem.pl.}}$ play$_{\text{fem.pl.}}$ in the garden is very beautiful$_{\text{fem.sing.}}$) is corrected in the singular (only two words corrected) rather than the plural (three words corrected).

To determine the score of an alternative x according to this criterion, we sum the occurrence of the pair (gender, number) with the occurrence of the tense and that of the personal pronoun. We obtain the following :

$f_1(x)$ = Occ(gender, number) + Occ(tense) + Occ(personal pronoun)

where Occ is the occurrence of the specified feature[3].

Let us consider the sentence presented above, the units concerned with the agreement and their features are represented in the following figure :



There are two possible corrections :

x_1= feminine, singular, present, 3

x_2= feminine, plural, present, 3

The outcomes of these alternatives are respectively :

$f_1(x_1) = Occ\ (F,S) = 3$

$f_1(x_2) = Occ\ (F,P) = 2$

Thus in terms of this criterion, we choose x_1 as the best correction since $f_1(x_1)$ is greater than $f_1(x_2)$.

3.2.2 Typographic criterion

This criterion is devoted to agreement errors which have a lexical origin. These errors can not be detected by the lexical analysis given they belong to the lexicon; however they are detected by the parser as agreement errors since they didn't fire the agreement rules. For instance, in the sentence : ' البنت الصغيرات تلعبن بدراجتهن في الحديقة ' (The little$_{fem.pl.}$ girl$_{fem.sing.}$ play$_{fem.pl.}$ with their$_{fem.pl.}$ bicycle in the garden), it is clear that the agreement error in number between the noun 'البنـت' (girl) and the other units of the phrase has a lexical origin. So, in order to write 'البنـات' (girls), the user can omit the letter 'ا' and then writes the word 'البنت' (girl) which belongs to the lexicon.

To carry on this kind of errors, the typographic criterion favors the correction which minimizes the typographic transformations of the erroneous words. In most cases these transformations can be either an omission of a letter (e.g., ' بنات' (girls) → ' بنت' (girl)), an

[3] If the specified feature doesn't fail in terms of the unification process, we attribute zero to its associated occurrence.

addition of a letter (e.g., 'طالب' (student)$_{masc.sing.}$ → 'طالبة' (student)$_{fem.sing.}$) or a substitution of a letter by another (e.g., 'مهندسون' (engineers)$_{masc.pl.}$ → 'مهندسان' (engineers)$_{masc.duel}$). The permutation errors occur in some cases and they are generally followed by one of the errors presented above (e.g., 'مهندسات' (engineers)$_{fem.pl.}$ → 'مهندستان' (engineers)$_{fem.duel}$: permutation of two letters and omission of a letter).

To take into account the transformations necessary to correct an erroneous word, we assign the following ordinal scores to the different kind of errors (Ben Hamadou 93) :

- score of a letter omission ($W_o = 0.25$)
- score of substitution of a letter by another ($W_s = 0.5$)
- score of permutation of two letters ($W_p = 0.75$)
- score of a letter addition ($W_a = 1$)

These scores are chosen according to the frequency of each kind of error. For instance, the writer can omit a letter where it is necessary, but rarely adds one where it is not. Consequently, the score of omission of a letter is strictly lower than that of the addition.

The outcome $f_2(x)$ of an alternative x in terms of this criterion is the sum of the scores of the different typographic transformations which affect the erroneous words. For instance, to change the word 'طالب' (student$_{masc.sing.}$) by the word 'طلبة' (students $_{masc.pl.}$) there is a total score of 1.25 (addition of a letter and omission of a letter).

3.2.3 Morphological criterion

Generally, the correction of an erroneous word requires the change of some of its letters. Thus, in English, to conjugate a verb in the present with a plural personal pronoun, we omit the letter 's' from the singular form (e.g., he eats → they eat). However, in French, we must add the letters 'ent' (the plural mark) (e.g., il mange $_{masc. sing.}$ → ils mangent $_{masc. pl.}$). The same thing is used for Arabic since we add the plural mark which is generally represented by one or many letters (e.g., هو يأكل → هم يأكلون).

We can say that the change of the tense or the personal pronoun of a verb requires the addition/omission of some of its letters. This is not usually the case for the nouns and adjectives since many words have restricted morphological features. For example, the noun 'رجل' (man) can be only masculine and the adjective 'حامل' (pregnant) can be only feminine. In these cases, to change the morphological features of a word, we must change the root of the

word. For instance, the masculine form of the word 'بنت' (girl) is 'ولد' (boy) (the second word is not derived from the first one since they didn't share the same root).

To handle this kind of errors, the morphological criterion favors the corrections that do not change the root of a word. Consider the sentence : 'البنت ياكلون تفاحة' (The $girl_{fem.sing.}$ $eat_{masc.pl.}$ an apple) which includes an agreement error in gender between the subject ' بنت' (girl) and the verb 'ياكلون' (eat). The best alternative in terms of this criterion is the feminine-singular: it is more simple to correct the verb by changing some of its letters (a substitution of a letter by another and a deletion of two words) than to change the noun by another which is not derived from it. The idea here is that the user may omit or add some letters by mistake rather replaces a word by another.

This criterion, which we have to minimize, is measured by the occurrence of such words in the sentence. Each alternative x is evaluated by a score $f_3(x)$ that represents the number of words with restricted morphological features to be corrected.

4 Criteria aggregation by the TOPSIS method

4.1 Main steps of TOPSIS

TOPSIS is a multiple criteria decision making method (MCDM) devoted to cardinal criteria.

According to this method, the best solutions are defined to be those which are farthest from the negative-ideal point (the alternative with worst scores on all criteria) as well as closest to the ideal point (the alternative which has the best scores on all criteria). The ideal point and the negative-ideal point can be two artificial (not feasible) alternatives.

The various steps of TOPSIS may be summarised as follows ((Yoon and Hwang 81), (Hwang and Yoon 85)) :

step0: Construction of the decision matrix

Let $Y=(y_{ij})$, $i = 1 \ldots n$, $j = 1 \ldots q$ be the decision matrix such that $y_{ij} = f_j(x_i)$.

y_{ij} is the outcome of the alternative x_i with respect to the criterion fj and Y represents the outcomes of each alternative in terms of all criteria.

step1: Construction of the normalised decision matrix

This step tries to transform the various attribute dimensions into non-dimensional attribute in order to allow comparison across the attributes.

The corresponding element of the normalised decision matrix can be calculated as :

$$r_{ij} = \frac{y_{ij}}{\left\{\sum_{i=1}^{n} y_{ij}^2\right\}^{1/2}} \qquad i = 1,\ldots,n \qquad j = 1,\ldots,q$$

step2: Construction of the weighted normalised decision matrix

This matrix is obtained by multiplying each column of the normalised decision matrix with its associated weight (α_j). An element of the new matrix will be :

$$v_{ij} = \alpha_j r_{ij} \qquad i = 1,\ldots,n; \quad j = 1,\ldots,q$$

step3: Determination of ideal and negative-ideal solutions

The ideal solution (x^+) is defined as :

$$x^+ = \left\{v_1^+, v_2^+, \ldots, v_j^+, \ldots, v_q^+\right\}$$

$$\text{where } v_j^+ = \left\{\max_i v_{ij}, j \in J, \min_i v_{ij} \quad j \in J'\right\}$$

J is the set of criteria to be maximized (frequency criterion) and J' is the set of criteria to be minimized (typographic criterion and morphological criterion).

The negative-ideal solution (x^-) is defined as :

$$x^- = \left\{v_1^-, v_2^-, \ldots, v_j^-, \ldots, v_q^-\right\}$$

$$\text{where } v_j^- = \left\{\min_i v_{ij}\, j \in J, \max_i v_{ij} \quad j \in J'\right\}$$

step4: Calculation of the separation measure

This step tries to measure the separation (in terms of Euclidean distance) of each alternative from the ideal solution as follows:

$$S_i^+ = \left\{\sum_{j=1}^{q} (v_{ij} - v_j^+)^2\right\}^{1/2} \qquad i = 1,\ldots,n$$

Similarly, the separation from the negative-ideal solution is given by :

$$S_i^- = \left\{\sum_{j=1}^{q} (v_{ij} - v_j^-)^2\right\}^{1/2} \qquad i = 1,\ldots,n$$

step5: Calculation of the relative closeness to the ideal solution

The relative closeness of an alternative x_i with respect to x^+ is defined by :

$$C_i^+ = \frac{S_i^-}{\left(S_i^+ + S_i^-\right)} \qquad 0 < C_i^+ < 1 \qquad i = 1, \ldots, n$$

Then the preference order can be obtained according to the descending order of C_i^+ and the best alternative will be defined as the one which is closer to x^+ than to x^-.

4.2 Weighting the different criteria

In the following we present subjective weighing criteria experimented on a variety of real sentences.

Determination of the weight of the frequency criterion (α_1)

The frequency criterion is more important when the difference in terms of score between the best alternative and the other ones is very important.

Then, α_1 may be defined by :

$$\alpha_1 = \sum_{i=1}^{n} \left(\operatorname{Max}_i f_1(x_i) - f_1(x_i)\right)$$

Determination of the weight of the morphological criterion (α_2)

Our experimental study of test sentences shows that α_2 depends on α_1 if ($\alpha_1 \neq 0$) otherwise it depends on the frequency of the alternatives :

If $\operatorname{Max} f_1(x_i) > 5 \operatorname{Max} f_2(x_i)$ then α_1 is more important than α_2.

If $\operatorname{Max} f_1(x_i) \leq 5 \operatorname{Max} f_2(x_i)$ then α_2 is more important than α_1.

α_2 may be defined as follows :

if $\alpha_1 \neq 0$ then

$$\alpha_2 = \begin{cases} \frac{1}{5}\alpha_1 & \text{if } \operatorname{Max} f_1(x_i) > 5 \operatorname{Max} f_2(x_i) \\ \frac{1}{\alpha_1} & \text{if } \operatorname{Max} f_1(x_i) \leq 5 \operatorname{Max} f_2(x_i) \end{cases}$$

if $\alpha_1 = 0$ then

$$\alpha_2 = \begin{cases} \frac{1}{5} \operatorname{Max} f_1(x_i) & \text{if } \operatorname{Max} f_1(x_i) > 5 \operatorname{Max} f_2(x_i) \\ \frac{1}{\operatorname{Max} f_1(x_i)} & \text{if } \operatorname{Max} f_1(x_i) \leq 5 \operatorname{Max} f_2(x_i) \end{cases}$$

Note that (1/5) represents the trade-off between the morphological criterion and the frequency criterion.

Determination of the weight of the typographic criterion (α_3)

α_3 depends on the typographic gaps ($\sum_{i=1}^{n} (\underset{i}{\text{Max}}\, f_3(x_i) - f_3(x_i))$) and conversely depends on the frequency gaps ($\sum_{i=1}^{n} (\underset{i}{\text{Max}}\, f_1(x_i) - f_1(x_i))$). α_3 may be done by :

$$\alpha_3 = \begin{cases} \dfrac{\sum_{i=1}^{n} (\underset{i}{\text{Max}}\, f_3(x_i) - f_3(x_i))}{\alpha_1} & \text{If } \alpha_1 \neq 0 \\ \dfrac{\sum_{i=1}^{n} (\underset{i}{\text{Max}}\, f_3(x_i) - f_3(x_i))}{\underset{i}{\text{Max}}\, f_1(x_i)} & \text{If } \alpha_1 = 0 \end{cases}$$

Note that the weights (α_1 , α_2 , α_3) are calculated on the basis of the normalised decision matrix and since they must satisfy the constraints :

$0 \leq \alpha_j \leq 1$ and $\sum_{j=1}^{q} \alpha_j = 1$, we will normalise them by : $\alpha_j = \dfrac{\alpha_j}{\sum_{j=1}^{q} \alpha_j}$

4.3 An illustrative example

Let us consider the sentence : 'هذا الرجل الغنية إستأجر حانوتين ووضعا متاعهما فيهما ' (this $_{\text{masc.sing.}}$ rich $_{\text{fem.sing.}}$ man $_{\text{masc.sing.}}$ rented$_{\text{masc.sing.}}$ two shops and put $_{\text{masc.duel}}$ their $_{\text{masc.duel}}$ goods in them)

The result of the "Extended Syntagmatic Analysis" is given by the following extended syntagms :

SE_1= { هما (their), وضعا (put), إستأجر (rented), الغنية (rich), الرجل (man), هذا (this) }

SE_2= { هما (them), حانوتين (two shops) }

Let us consider SE_1. The features of SE_1 may be represented by the following figure which shows the errors in gender and number.

	هذا This	الرجل man	الغنية rich	إستأجر rented	وضعا put	هما their	
Gender	M	M	F	M	M	M	→ Error in gender
Number	S	S	S	S	D	D	→ Error in number
Tense	X	X	X	P	P	X	
Presonal pronoun	3	3	3	3	3	3	

Clearly, there are three correction alternatives :

x_1: Masculine, singular

x_2: Masculine, duel

x_3: Feminine singular

The scoring of these alternatives in terms of the criteria is given in the following decision matrix :

	x_1	x_2	x_3
Frequency Criterion	3	2	1
Morphological Criterion	0	0	1
Typographic Criterion	4	1,75	7,25

According to this matrix, we can conclude that x_3 is dominated by x_1 and x_2. x_1 and x_2 are two efficient solutions.

step1: Construction of the normalised decision matrix

	x_1	x_2	x_3
Frequency Criterion	0,80	0,534	0,267
Morphological Criterion	0	0	1
Typographic Criterion	0,472	0,206	0,856

step2: Construction of the weighted normalised decision matrix

The weights of each criterion are respectively :

$$\alpha_1 = (0{,}8 - 0{,}53) + (0{,}8 - 0{,}267) = 0{,}8 \qquad \alpha_1 = 0{,}239$$

$$\alpha_2 = \frac{1}{0{,}8} = 1{,}25 \xrightarrow{\text{Normalised weights}} \alpha_2 = 0{,}373$$

$$\alpha_3 = \frac{(0{,}856 - 0{,}472) + (0{,}856 - 0{,}206)}{0{,}8} = 1{,}29 \qquad \alpha_3 = 0{,}386$$

The normalised decision matrix is the following :

	x_1	x_2	x_3
Frequency Criterion	0,191	0,127	0,063
Morphological Criterion	0	0	0,373
Typographic Criterion	0,182	0,079	0,331

step3: Determination of ideal and negative-ideal solutions

Ideal solution V^+	***Anti-ideal solution*** V^-
0,191	0,063
0	0,373
0,079	0,331

step4: Calculation of the separation measure

	x_1	x_2	x_3
S+	0,102	0,063	0,468
S-	0,422	0,455	0

step5: Calculation of the relative closeness to the ideal solution

	S+	*S-*	*C+*	*Ranking*
x_1	0,102	0,422	0,804	2
x_2	0,063	0,455	0,876	1
x_3	0,468	0	0	3

x_2 has the best ranking, thus the best correction is obtained by lining up all the words of the erroneous sentence by the Masculine- Duel :

"هـذان الرجـلان الغنيـان إستأجرا حـانوتين ووضعـا متاعهمـا فيهمـا" (these$_{masc.\ Duel}$ rich $_{masc.\ Duel}$ men $_{masc.\ Duel}$ rented$_{masc.\ Duel}$ two shops and put $_{masc.Duel}$ their $_{masc.Duel}$ goods in them).

5 Preliminary Experiment

A prototype implementation of the proposed method called 'DECORA' is developed using the C++ programming language with WINDOWS environment.

In order to evaluate the DECORA performance, 300 sentences are chosen from real texts written by secondary school students. The sentences are various : they contain from 1 to 4 extended syntagms and each syntagm contains a maximum of 9 words.

The sentences are corrected by a human expert whom we ask to classify them in the three following classes :

Class 1 : sentences which corrections are obvious.

Class 2 : sentences which corrections are not obvious and are somewhat challenging.

Class 3 : sentences which corrections need a very careful analysis to choose among the possible ones.

The same sentences are processed by DECORA. The results of the comparison between DECORA and the expert corrections are given in the following table :

(A) \ (B)	1	2	≥3
1	100 %	0 %	0 %
2	85 %	13 %	1 %
3	74 %	21 %	5 %

(A) : Sentences classes

(B) :Ranks of the corrections proposed by DECORA which are similar to the expert corrections.

As shown in this table, for the first sentences class, all corrections proposed by DECORA as the best ones are similar to those given by the human expert. 85 % of sentences of the second class are corrected in the same way by DECORA and the expert. This percentage decreases to 74 % for the third class. DECORA's best proposed corrections of the reminding sentences are different from those of the expert. Generally, these sentences have more than one plausible correction (i.e., they have very close scores). The expert may in some cases be hesitating between two or more possible corrections and then the choice of the best one is almost made at random. However, DECORA can make distinction between these alternatives by ranking them according to their scores.

Note that in some cases of the third class, different human experts may have different opinions about the best correction. In fact they may disagree with the relative importance of each criteria.

We think that the obtained results are very satisfying and we hope to obtain better results by studying more real sentences in order to improve the criteria weights.

6 Conclusion

As the correction process of agreement errors is not usually a simple task since the choice of the best correction alternative requires a careful analysis, we think that the use of a multicriteria approach to guide the correction process is very interesting.

In this paper, we proposed three main criteria (the frequency criterion, the morphological criterion and the typographic criterion) to rank the correction alternatives of Arabic sentences. We presented the techniques of scoring the correction alternatives in terms of the considered criteria. Finally, we showed that using TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) as an aggregation method of the considered criteria is well appropriated for our problem and the obtained results are very satisfying.

Acknowledgements

The authors are grateful to Mr Foued Ben Abdelaziz : professor at the management school "Institut Superieur de Gestion de Tunis" for his interest to this work and his comments on the preliminary version of the paper.

References

BELGUITH L. and BEN HAMADOU A., "Un vérificateur et correcteur des accords pour l'arabe non voyellé", Actes de la 3ème conférence annuelle sur le traitement automatique du langage naturel, 22-24 Mai, Marseille, 1996.

BEN HAMADOU A., "Vérification et correction automatique par analyse affixale des textes écrits en langage naturel : le cas de l'arabe non voyellé", Thèse d'Etat en informatique, Faculté des Sciences de Tunis, 1993.

BEN HAMADOU A. and BELGITH L., "Une méthode de vérification et de correction des accords appliquée à l'arabe non voyellé", Actes de la 1ère conférence internationale NLP+IA, Moncton, Canada, 4-6 Juin 1996.

BLACHE P., "L'analyse par filtrage ascendant: Une stratégie efficace pour les grammaires syntagmatiques généralisées", Actes de la 10ème conférence internationale sur le traitement du langage naturel et ses applications, Avignon, 1990.

BLACHE P., "Problèmes d'analyse syntaxique pour les correcteurs grammaticaux", Actes de la 11ème conférence internationale sur le traitement du langage naturel et ses applications, Avignon, 1991.

BOLIOLI A., DINI L. and MALNATI G., "Parsing Italian with a robust constraint grammar", International Conference on Computational Linguistics, Coling 1992.

COCH J.and MORIZE G., "Un analyseur conçu pour le traitement d'erreurs et ambiguïtés", Actes de la 10ème conférence internationale sur le traitement du langage naturel et ses applications, Avignon, 1990.

GENTHIAL D., COURTIN J. and KOWARSKI I., "Contribution of a category hierarchy to the robustness of syntactic parsing", Coling' 90, pp. 139-144, Helsinki, 1990.

GENTHIAL D and COURTIN J., "From Detection / Correction to computer Aided Writing", Coling'92, pp. 1013-1018, Nantes, 1992.

GENTHIAL D., COURTIN J. and MENEZO J., "Towards a More User-Friendly Correction", CoLing'94, pp. 1083-1088, 1994.

HWANG C.L. and YOON K., "Multiple attribute decision Making- Methods and applications : A State of the art survey", Springer-Verlag, New York, 1981.

LAPALME G. and RICHARD D., "Un système de correction automatique des accords des participes passés", Technique et Science Informatique, N°4, 1986.

LAPALME G.and LACOURTE R., "Une implantation informatique du français fondamental", Technique et science Informatique, N° 5, 1988.

PO-LUNG YU, "Multiple-criteria Decision Making concepts, Techniques, and extensions", PLENUM Press, NEW YORK, 1985.

RAVIN Y., "Grammar errors and style weakness in a text-critiquing system", IEEE Transactions on Professional Communication, Vol. 31, N° 3, September 1988.

STRUBE DE LUMA V.L., "Contribution à l'étude du traitement des erreurs au niveau lexico-syntaxique dans un texte écrit en français", Thèse de doctorat, institut IMAG, Université JOSEPH FOURIER, 1990.

VERONIS J., "Morphosyntactic correction in natural language interfaces", 12th International Conference on Computational Linguistics Coling, 1988.

VERONIS J., "Error in natural language dialogue between man and machine", Internatonal Journal Man - Machine Studies, Vol. 35, pp. 187-217, 1991.

VOSSE T., "Detecting and correcting morpho-syntactic errors in real texts", Proceedings of the third conference on applied natural language processing, Trento Italy, 31 March - 3 April, pp. 111-118, New York ACL, 1992.

YOON K. and HWANG C., "Manufacturing plant location analysis by multiple attribute decision making", International journal of production research, Vol. 23, N°2, pp. 345-359, 1985.

Incorporating Bigram Constraints into an LR Table

Hiroki Imai, Hui Li and **Hozumi Tanaka**
Department of Computer Science, Tokyo Institute of Technology
2-12-1 O-okayama, Meguro, Tokyo 152 Japan
{imai,li,tanaka}@cs.titech.ac.jp

Abstract

In this paper, we propose a method to construct a bigram LR table to incorporate bigram constraints into an LR table. An LR table which incorporates bigram constraints is called a bigram LR table. Using the bigram LR table, it is possible for a GLR parser to make use of both bigram and CFG constraints in natural language processing.

A method for constructing bigram LR tables is proposed. Applying the resultant bigram LR table to our GLR method has the following advantages:

1. A language model utilizing a bigram LR table has lower perplexity than a bigram language model, since local constraints (bigram) and global constraints (CFG) are combined in the single bigram LR table at the same time.
2. Bigram constraints are easily acquired from a given corpus. Therefore data sparseness is not likely to arise.

The former advantage leads to a reduction in complexity, and as the result, produces better performance for GLR parsing.

Our experiments demonstrate the effectiveness of our method.

1 Introduction

In natural language processing, stochastic language models are commonly used for lexical and syntactic disambiguation (Fujisaki et al., 1991; Franz, 1996). Stochastic language models are also helpful in reducing the complexity of speech and language processing by way of providing probabilistic linguistic constraints (Lee, 1989).

N-gram models (Jelinek, 1990), including bigram and trigram models, are the most commonly used method of applying local probabilistic constraints. However, context-free grammars (CFGs) produce more global linguistic constraints than N-gram models. It seems better to combine both local and global constraints and use them both concurrently in natural language processing. The reason why N-gram models are preferred over CFGs is that N-gram constraints are easily acquired from a given corpus. However, the larger N is, the more serious the problem of data sparseness becomes.

CFGs are commonly employed in syntactic parsing as global linguistic constraints, since many efficient parsing algorithms are available. GLR (Generalized LR) is one such parsing algorithm that uses an LR table, into which CFG constraints are precompiled in advance (Knuth, 1965; Tomita, 1986).

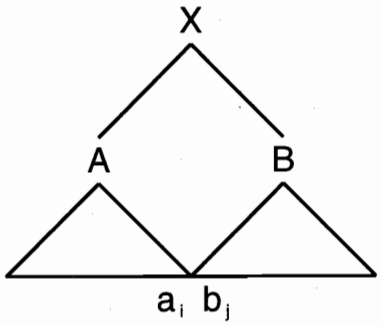


Figure 1: Connection check by CFG

Therefore if we can incorporate N-gram constraints into an LR table, we can make concurrent use of both local and global linguistic constraints in GLR parsing.

In the following section, we will propose a method that incorporates bigram constraints into an LR table. The advantages of the method are summarized as follows:

First, it is expected that this method produces a lower perplexity than that for a bigram language model, since it is possible to utilize both local (bigram) and global (CFG) constraints in the LR table. We will evidence this reduction in perplexity by considering states in LR table for the case of GLR parsing.

Secondly, bigram constraints are easily acquired from smaller-sized corpora. Accordingly, data sparseness is not likely to arise.

2 CFG, Connection Matrix and LR table

2.1 Relation between CFG and Connection Constraints

Figure 1 represents a situation in which a_i and b_j are adjacent each other, where a_i belongs to Set_I $(i = 1, \cdots, I)$ and b_j belongs to Set_J $(j = 1, \cdots, J)$. Set_I and Set_J are defined by $last1(A)$ and $first1(B)$(Aho et al., 1986), respectively. If $a \in Set_I$ and $b \in Set_J$ happen not to be able to occur in this order, it becomes a non-trivial task to express this adjacency restriction within the framework of CFG.

One solution to this problem is to introduce a new nonterminal symbol A_i for each a_i and a nonterminal symbol B_j for each b_j. Introducing new nonterminal symbols A_i and B_j, we replaces the rule $X \rightarrow A\ B$ with a set of rules of $\{X \rightarrow A_i\ B_j \mid$ for all pairs (A_i, B_j) where b_j can follow $a_i\}$. After this rule replacement, the order of the number of rules will become $I \times J$ in the worst case. The introduction of such new nonterminal symbols leads to an increase in grammar rules, which not only makes the LR table very large in size, but also diminishes efficiency of the GLR parsing method.

The second solution is to augment $X \rightarrow A\ B$ with a procedure that checks the connection between a_i and b_j. This solution can avoid the problem of the expansion of CFG rules, but we have to take care

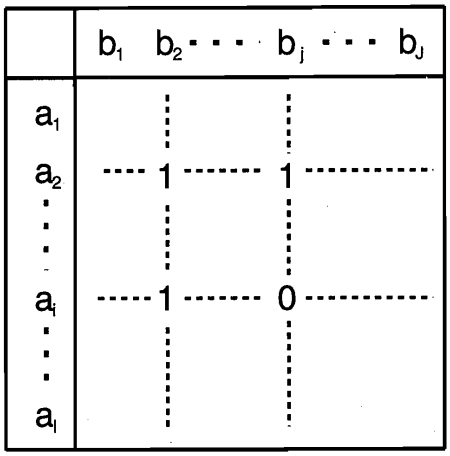


Figure 2: Connection matrix

of the information flow from the bottom leaves to the upper nodes in the tree, A, B, and X.

Neither the first nor the second solution are preferable, in terms of both efficiency of GLR parsing and description of CFG rules. Additionally, it is a much easier task to describe local connection constraints between adjacent two terminal symbols by way of a connection matrix such as in Figure 2, than to express these constraints within the CFG.

The connection matrix in Figure 2 is defined as:

$$Connect(a_i, b_j) = \begin{cases} 1 & \text{if } b_j \text{ can follow } a_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The best solution seems to be to develop a method that can combine both a CFG and a connection matrix, avoiding the expansion of CFG rules. Consequently, the size of the LR table will become smaller and we will get better GLR parsing performance. In the following section, we will propose one such method.

2.2 Relation between the LR Table and Connection Matrix

First we discuss the relation between the LR table and a connection matrix. The action part of an LR table consists of lookahead symbols and states. Let a shift action *sh* m be in state l with the lookahead symbol a. After the GLR parser executes action *sh* m, the symbol a is pushed onto the top of the stack and the GLR parser shifts to the state m. Suppose there is an action A in state m with lookahead b (see Figure 3). The action A is executable if $Connect(a, b) \neq 0$ (b can follow a), whereas if $Connect(a, b) = 0$ (b cannot follow a), the action A in state m with lookahead b is not executable and we can remove it from the LR table as an invalid action. Removing such invalid actions enables us to incorporate connection constraints into the LR table in addition to the implicit CFG constraints.

In section 3.2, we will propose a method that integrates both bigram and CFG constraints into an LR table. After this integration process, we obtain a table called a bigram LR table.

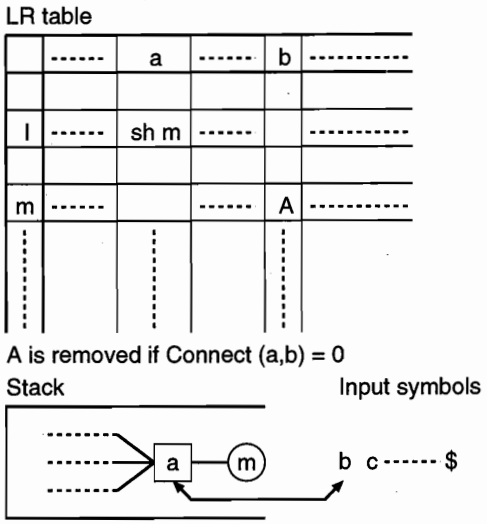


Figure 3: LR table and Connection Constraints

3 Integration of Bigram and CFG Constraints into an LR Table

3.1 The Definition of a Probabilistic Connection Matrix

A close relation exists between bigrams and connection matrices, in that the bigram probability $P(b|a)$ corresponds to the matrix element of $Connect(a, b)$. A connection matrix incorporating bigram probabilities is called a probabilistic connection matrix, in which $Connect(a, b) = 0$ still means b cannot follow a, but instead of connection matrix entries having a binary value of 0 or 1, a probability is associated with each element. This is then used to construct a probabilistic LR table.

The N-gram model is the most commonly used probabilistic language model, and it assumes that a symbol sequence can be described by a higher order Markov process. The simplest N-gram model with $N = 2$ is called a bigram model, and approximates the probability of a string $\mathbf{X} = x_1 x_2 x_3 \cdots x_n$ as the product of conditional probabilities:

$$P(\mathbf{X}) = P(x_1|\#)P(x_2|x_1)\cdots P(x_n|x_{n-1})P(\$|x_n) \tag{2}$$

In the above expression, "#" indicates the sentence beginning marker and "$" indicates the sentence ending marker. The above bigram model can be represented in a *probabilistic connection matrix* defined as follows.

DEFINITION 1 (*probabilistic connection matrix*)

Let $G = (V_N, V_T, P, S)$ be a context-free grammar. For $\forall a, b \in V_T$ (the set of terminal symbols), the *probabilistic connection matrix* named *PConnect* is defined as follows.

$$PConnect(a, b) = P(b|a) \tag{3}$$

where $P(b|a)$ is a conditional probability and $\sum_{b \in V_T} P(b|a) = 1$.

$PConnect(a, b) = 0$ means that a and b cannot occur consecutively in the given order. $PConnect(a, b) \neq 0$ means b can follow a with probability $P(b|a)$.

3.2 An algorithm to construct an bigram LR table

An algorithm to construct a probabilistic LR table, combining both bigram and CFG constraints, is given in Algorithm 1:

Algorithm 1

Input: A CFG $G = (V_N, V_T, P, S)$ and a probabilistic connection matrix *PConnect*.

Output: An LR table T with CFG and bigram constraints.

Method:

Step 1 Generate an LR table T_0 from the given CFG G.

Step 2 Removal of actions:

For each shift action *sh* m with lookahead a in the LR table T_0, delete actions in the state m with lookahead b if $PConnect(a, b) = 0$.

Step 3 Constraint Propagation (Tanaka et al., 1994):

Repeat the following two procedures until no further actions can be removed:

1. Remove actions which have no succeeding action,
2. Remove actions which have no preceding action.

Step 4 Compact the LR table if possible.

Step 5 Incorporation of bigram constraints into the LR table:

For each shift action *sh* m with lookahead a in the LR table T_0, let

$$P = \sum_{i=1}^{N} PConnect(a, b_i)$$

where $\{b_i : i = 1, \cdots, N\}$ is the set of lookaheads for state m. For each action A_j in state m with lookahead b_i, assign a probability p to action A_j:

$$p = \frac{P(b_i|a)}{P \times n} = \frac{PConnect(a, b_i)}{P \times n}$$

where n is the number of conflict actions in state m with lookahead b_i. The denominator is clearly a normalization factor.

Step 6 For each shift action A with lookahead a in state 0, assign A a probability $p = P(a|\#)$, where "#" is the sentence beginning marker.

(1)	$S \rightarrow X\ Y$	(6)	$A \rightarrow a1$
(2)	$X \rightarrow A$	(7)	$A \rightarrow a2$
(3)	$X \rightarrow A\ B$	(8)	$B \rightarrow b1$
(4)	$Y \rightarrow A$	(9)	$B \rightarrow b2$
(5)	$Y \rightarrow b1\ A$		

Figure 4: Grammar G_1

	a1	*a2*	*b1*	*b2*	$
#	*0.6*	*0.4*	*0.0*	*0.0*	*0.0*
a1	*0.0*	*0.0*	*0.0*	*1.0*	*0.0*
a2	*0.0*	*0.0*	*0.3*	*0.0*	*0.7*
b1	*0.0*	*0.1*	*0.9*	*0.0*	*0.0*
b2	*0.0*	*0.0*	*1.0*	*0.0*	*0.0*

Figure 5: Probabilistic connection matrix M_1

Step 7 Assign a probability $p = 1/n$ to each action A in state m with lookahead symbol a that has not been assigned a probability, where n is the number of conflict actions in state m with lookahead symbol a.

Step 8 Return the LR table T produced at the completion of Step 7 as the *Bigram LR table.*

As explained above, the removal of actions at Step 2 corresponds to the operation of incorporating connection constraints into an LR table. We call Step 3 Constraint Propagation which reduces the size of the LR table (Li, 1996). As many actions are removed from the LR table during Step 2 and 3, it becomes possible to compress the LR table in Step 4. We will demonstrate one of such example in the following section.

It should be noted that the above algorithm can be applied to any type of LR table, that is a canonical LR table, an LALR table, or an SLR table.

4 An Example

4.1 Generating a Bigram LR Table

In this section, we will provide a simple example of the generation of a bigram LR table by way of applying Algorithm 1 to both a CFG and a probabilistic connection matrix, to create a bigram LR table. Figure 4 and Figure 5 give a sample CFG G_1 and a probabilistic connection matrix M_1, respectively.

Note that grammar G_1 in Figure 4 does not explicitly express local connection constraints between terminal symbols. Such local connection constraints are easily expressed by a matrix M_1 as shown in Figure 5.

From the CFG given in Figure 4, we can generate an LR table, Table 1, in Step 1 using the conventional LR table generation algorithm.

state	**action**					**goto**				
	a1	*a2*	*b1*	*b2*	*$*	*A*	*B*	*X*	*Y*	*S*
0	*sh1*	*sh2*				*3*		*4*		*5*
1	*re6*	*re6*	*re6*	*re6*						
2	*re7*	*re7*	*re7*	*re7*						
3	*re2*	*re2*	*re2/sh6*	*sh7*			*8*			
4	*sh9*	*sh10*	*sh11*			*12*			*13*	
5					*acc*					
6	*re8*	*re8*	*re8*							
7	*re9*	*re9*	*re9*							
8	*re3*	*re3*	*re3*							
9					*re6*					
10					*re7*					
11	*sh9*	*sh10*				*14*				
12					*re4*					
13					*re1*					
14					*re5*					

Table 1: Initial LR table for G_1

state	**action**					**goto**				
	a1	*a2*	*b1*	*b2*	*$*	*A*	*B*	*X*	*Y*	*S*
0	*sh1*	*sh2*				*3*		*4*		*5*
1	*re6(2)*	*re6(2)*	*re6(2)*	*re6*						
2	*re7(2)*	*re7(2)*	*re7*	*re7(2)*						
3	*re2(3)*	*re2*	*re2/sh6*	*sh7*			*8*			
4	*sh9(3)*	*sh10*	*sh11*			*12*			*13*	
5					*acc*					
6	*re8(2)*	*re8*	*re8*							
7	*re9(3)*	*re9(2)*	*re9*							
8	*re3(3)*	*re3*	*re3*							
9					*re6(2)*					
10					*re7*					
11	*sh9(3)*	*sh10*				*14*				
12					*re4*					
13					*re1*					
14					*re5*					

Table 2: LR table after Step 2 and 3

Table 2 is the resultant LR table at the completion of Step 2 and Step 3, produced based on Table 1. Actions numbered *(2)* and *(3)* in Table 2 are those which are removed by Step 2 and Step 3, respectively.

In state 1 with a lookahead symbol *b1*, *re6* is carried out after executing action *sh1* in state 0, pushing *a1* onto the stack. Note that *a1* and *b1* are now consecutive, in this order. However, the probabilistic connection matrix (see Figure 5) does not allow such a sequence of terminal symbols, since $PConnect(a1, b1) = 0$. Therefore, the action *re6* in state 1 with lookahead *b1* is removed from Table 1 in Step 2, and thus marked as *(2)* in Table 2. For this same reason, the other *re6*s in state 1 with lookahead symbols *a1* and *a2* are also removed from Table 1.

On the other hand, in case of *re6* in state 1 with lookahead symbol *b2*, as *a1* can be followed by *b2* ($PConnect(a1, b2) \neq 0$), action *re6* cannot be removed. The reason remaining actions marked as *(2)* in Table 2 should be self-evident to the readers.

Next, we would like to consider the reason why action *sh9* in state 4 with lookahead *a1* is removed from Table 1. In state 9, *re6* with lookahead symbol \$ has already been removed in Step 2, and there is no succeeding action for *sh9*. Therefore, action *sh9* in state 3 is removed in Step 3, and hence marked as *(3)*.

Let us consider action *re3* in state 8 with lookahead *a1*. After this action is carried out, the GLR parser goes to state 4 after pushing X onto the stack. However, *sh9* in state 4 with lookahead *a1* has already been removed, and there is no succeeding action for *re3*. As a result, *re3* in state 8 with lookahead symbol *a1* is removed in Step 3. Similarly, *re9* in state 7 with lookahead symbol *a1* is also removed in Step 3. In this way, the removal of actions propagates to other removals. This chain of removals is called Constraint Propagation, and occurs in Step 3. Actions removed in Step 3 are marked as *(3)* in Table 2.

Careful readers will notice that there is now no action in state 9 and that it is possible to delete this state in Step 4. Table 3 shows the LR table after Step 4.

As a final step, we would like to assign bigram constraints to each action in Table 3. Let us consider the two *re8*s in state 6, reached after executing *sh6* in state 4 by pushing a lookahead of *b1* onto the stack. In state 6, P is calculated at Step 5 as shown below:

$$\begin{aligned} P &= PConnect(b1, a2) + PConnect(b1, b1) \\ &= 0.1 + 0.9 \\ &= 1 \end{aligned}$$

We can assign the following probabilities p to each *re8* in state 6 by way of Step 5:

$$p = \begin{cases} \dfrac{PConnect(b1, a2)}{P \times n} = \dfrac{0.1}{1 \times 1} = 0.1 & \text{for } re8 \text{ with lookahead } a2 \\ \dfrac{PConnect(b1, b1)}{P \times n} = \dfrac{0.9}{1 \times 1} = 0.9 & \text{for } re8 \text{ with lookahead } b1 \end{cases}$$

state	**action**					**goto**				
	a1	*a2*	*b1*	*b2*	*$*	*A*	*B*	*X*	*Y*	*S*
0	*sh1*	*sh2*				*3*		*4*		*5*
1				*re6*						
2			*re7*							
3		*re2*	*re2/sh6*	*sh7*			*8*			
4		*sh10*	*sh11*			*12*			*13*	
5					*acc*					
6		*re8*	*re8*							
7			*re9*							
8		*re3*	*re3*							
10					*re7*					
11		*sh10*				*14*				
12					*re4*					
13					*re1*					
14					*re5*					

Table 3: LR table after Step 4

After assigning a probability to each action in the LR table at Step 5, there remain actions without probabilities. For example, the two conflict actions (*re2/sh6*) in state 3 with lookahead *b1* are not assigned a probability. Therefore, each of these actions is assigned the same probability, 0.5, in Step 7. A probability of 1 is assigned to remaining actions, since there is no conflict among them.

Table 4 shows the final results of applying Algorithm 1 to G_1 and M_1.

4.2 Comparison of Language Models

Using the bigram LR table as shown in Table 4, the probability $P1$ of the string "*a2 b1 a2*" is calculated as:

$$\begin{aligned} P1 &= P(a2\ b1\ a2) \\ &= P(0, a2, sh2) \times P(2, b1, re7) \times P(3, b1, re2) \times P(4, b1, sh11) \\ &\quad \times P(11, a2, sh10) \times P(10, \$, re7) \times P(14, \$, re5) \times P(13, \$, re1) \\ &\quad \times P(5, \$, acc) \\ &= 0.4 \times 1.0 \times 0.5 \times 1.0 \times 1.0 \times 1.0 \times 1.0 \times 1.0 \times 1.0 \\ &= 0.2 \end{aligned}$$

where $P(S, L, A)$ means the probability of an action A in state S with lookahead L.

On the other hand, using only bigram constraints, the probability $P2$ of the string "*a2 b1 a2*" is calculated as:

$$\begin{aligned} P2 &= P(a2\ b1\ a2) \\ &= P(a2|\#) \times P(b1|a2) \times P(a2|b1) \times P(\$|a2) \\ &= \times 0.3 \times 0.1 \times 0.7 \end{aligned}$$

state	**action**					**goto**				
	a1	*a2*	*b1*	*b2*	*$*	*A*	*B*	*X*	*Y*	*S*
0	*sh1* *0.6*	*sh2* *0.4*				*3*		*4*		*5*
1				*re6* *1.0*						
2			*re7* *1.0*							
3		*re2* *1.0*	*re2/sh6* *0.5/0.5*	*sh7* *1.0*			*8*			
4		*sh10* *1.0*	*sh11* *1.0*			*12*			*13*	
5					*acc* *1.0*					
6		*re8* *0.1*	*re8* *0.9*							
7			*re9* *1.0*							
8		*re3* *1.0*	*re3* *1.0*							
10					*re7* *1.0*					
11		*sh10* *1.0*				*14*				
12					*re4* *1.0*					
13					*re1* *1.0*					
14					*re5* *1.0*					

Table 4: The Bigram LR table constructed by Algorithm 1

$$= \quad 0.0084$$

The reason why $P1 > P2$ can be explained as follows. Consider the beginning symbol *a2* of a sentence. In the case of the bigram model, *a2* can only be followed by either of the two symbols *b1* and \$ (see Figure 5). However, consulting the bigram LR table reveals that in state 0 with lookahead *a2*, *sh2* is carried out, entering state 2. State 2 has only one action *re7* with lookahead symbol *b1*. In other words, in state 2, \$ is not predicted as a succeeding symbol of *a1*. The exclusion of an ungrammatical prediction in \$ makes $P1$ larger than $P2$.

Perplexity is a measure of the complexity of a language model. The larger the probability of the language model is, the smaller the perplexity of the language model is. The above result ($P1 > P2$) indicates that the bigram LR table model gives smaller perplexity than the bigram model. In the next section, we will demonstrate this fact.

5 Evaluation of Perplexity

Perplexity is a measure of the constraint imposed by the language model. *Test-set perplexity* (Jelinek, 1990) is commonly used to measure the perplexity of a language model from a test-set. *Test-set perplexity* for a language model L is simply the geometric mean of probabilities defined by:

$$Q(L) = 2^{H(L)}$$

where

$$H(L) = \frac{1}{N} \sum_{i=1}^{M} \log P(S_i)$$

Here N is the number of terminal symbols in the test set, M is the number of test sentences and $P(S_i)$ is the probability of generating i-th test sentence S_i.

In the case of the bigram model, $P(S_i)$ is:

$$\begin{aligned} P(S_i) &= P(x_1, x_2, \cdots, x_n) \\ &= P(x_1|\#)P(x_2|x_1) \cdots P(x_n|x_{n-1})P(\$|x_n) \end{aligned}$$

Table 5 shows the *test-set perplexity* of allophones for each language model. Here the allophone bigram models (i.e. probabilistic allophone connection matrix) were trained on a corpus with about 220,000 phrases, with the open test-set consisting of about 17,000 phrases. The CFG used is a phrase context-free grammar used in speech recognition tasks, and the number of rules and words is 2813 and 1588, respectively.

As is evident from Table 5, the use of a bigram LR table decreases the *test-set perplexity* from 8.50 to 5.06, not considering CFG constraints, and from 4.30 to 2.95, with CFG constraints. This result shows the effectiveness of using a bigram LR table.

Even though the experiment described above is concerned with speech recognition, our method is applicable to all kinds of natural language processing systems.

Language model	Perplexity
Connection matrix	8.50
Bigram	5.06
CFG + Connection matrix	4.30
CFG + Probabilistic connection matrix (Bigram LR table)	2.95

Table 5: Perplexity of language models

6 Conclusions

In this paper, we described a method to construct a bigram LR table, and then discussed the advantage of our method, comparing our method to the bigram language model. The principle advantage is that, in using a bigram LR table, we can combine both local probabilistic connection constraints (bigram constraints) and global constraints (CFG).

It is well known that the perplexity of a bigram language model is greater than that of a trigram language model. We have already shown that the perplexity of a bigram language model is greater than that of a language model using a bigram LR table. It is an interesting question as to which of a trigram language model and a bigram LR table language model has larger perplexity. With regard to data sparseness, a bigram LR table language model is better than a trigram language model, since bigram constraints are easier to acquire than trigram constraints. In order to compare the bigram LR table language model with the trigram language model, we need to carry out further experimentation.

Su et al. (Su et al., 1991) and Chiang et al. (Chiang et al., 1995) have proposed a very interesting corpus-based natural language processing method that takes account not only of lexical, syntactic, and semantic scores concurrently, but also context-sensitivity in the language model. However, their method seems to suffer from difficulty in acquiring probabilities from a given corpus.

Wright (Wright, 1990) developed a method of distributing the probability of each PCFG rule to each action in an LR table. However, this method only calculates syntactic scores of parsing trees based on a context-free framework.

Briscoe and Carroll (Briscoe and Carroll., 1993) attempt to incorporate probabilities into an LR table. They insist that the resultant probabilistic LR table can include probabilities with context-sensitivity. In a recent technical report, (Inui et al., 1997) reported out that the resultant probabilistic LR table has a defect in terms of the process used to normalize probabilities associated with each action in the LR table. Inui et. al. are now obtaining promising experimental results which will be published elsewhere.

Finally, we would like to mention that Klavans and Resnik (Klavans and Resnik, 1996) have advocated a similar approach to ours which combines symbolic and statistical constraints, CFG and bigram constraints.

References

A.V. Aho, S. Ravi, and J.D. Ullman. 1986. *Compilers: Principle, Techniques, and Tools.* Addison Wesley.

T. Briscoe and J. Carroll. 1993. Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1):25–59.

T.H. Chiang, Y.C. Lin, and K.Y. Su. 1995. Robust learning, smoothing, and parameter tying on syntactic ambiguity resolution. *Computational Linguistics*, 21(3):321–349.

A. Franz. 1996. *Automatic Ambiguity Resolution in Natural Language Processing.* Springer.

T. Fujisaki, F. Jelinek, J. Cocke, E. Black, and T. Nishino. 1991. A probabilistic parsing method for sentence disambiguation. In M. Tomita, editor, *Current Issues in Parsing Technologies*, pages 139–152. Kluwer Academic Publishers.

K. Inui, V. Sornlertlamvanich, H. Tanaka, and T. Tokunaga. 1997. A new probabilistic LR language model for statistical parsing. Technical Report TR97-0004, Department of Computer Science, Tokyo Institute of Technology.

F. Jelinek. 1990. Self-organized language modeling for speech recognition. In A. Waibel and K.F. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann.

J.L. Klavans and P. Resnik. 1996. *The Balancing Act: Combining Symbolic and Statistial Approaches to Language.* The MIT Press.

D.E. Knuth. 1965. On the translation of languages left to right. *Information and Control*, 8(6):607–639.

K.F. Lee. 1989. *Automatic Speech Recognition: The Development of the SPHINX System.* Kluwer Academic Publishers.

H. Li. 1996. Incorporation of phoneme-context-depencence into LR table through constraint propagation method. *Journal of Japanese Society for Artificial Intelligence*, 11(2):246–254.

K.Y. Su, J.N. Wang, M.H. Su, and J.S. Chang. 1991. GLR parsing with scoring. In M. Tomita, editor, *Generalized LR Parsing.* Kluwer Academic Publishers.

H. Tanaka, H. Li, and T. Tokunaga. 1994. Incorporation of phoneme-context-dependence into LR table through constraints propagation method. In *Workshop on Integration of Natural Language and Speech Processing*, pages 15–22.

M. Tomita. 1986. *Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems.* Kluwer Academic Publishers.

J.H. Wright. 1990. LR parsing of probabilistic grammars with input uncertainty for speech recognition. *Computer Speech and Language*, 4(4):297–323.

A Level-synchronous Approach to Ill-formed Sentence Parsing

Yi-Chung Lin* and Keh-Yih Su†

* Advanced Technology Center, Computer and Communication Research Lab.,
Industrial Technology Research Institute, Hsinchu, Taiwan 310, R.O.C.
lyc@e0sun3.ccl.irti.org.tw

† Department of Electrical Engineering, National Tsing-Hua University
Hsinchu, Taiwan 300, R.O.C.
kysu@bdc.com.tw

Abstract

In this paper, a *Phrase-Level-Building* (PLB) mechanism is proposed to parse ill-formed sentences. By decomposing a syntactic tree into phrase-levels, this mechanism regards the task of parsing a sentence as a task of building the phrase-levels for the sentence. During parsing, a level-synchronous scoring function is used to remove less likely phrase-levels. As a result, instead of enumerating all possible parses, the PLB parser only generates the more likely *tree groups*, each of which is a set of partial parses jointly deriving the input. Whenever all active phrase-levels in the search beam cannot be further reduced by any grammar rules, the process of building phrase-levels is stopped and a probabilistic scoring function is used to select the best tree group. With this approach, the best tree group is selected within a wider scope (i.e., the whole sentence), and thus generates better result. Compared with the baseline system using the stochastic context-free grammar and the "leftmost longest phrase first" heuristics (which operates in a narrow scope), the proposed PLB approach improves the precision of brackets in the tree group from 69.37% to 79.49%. The recall of brackets is also improved from 78.73% to 81.39%.

1 Introduction

Natural language parsing plays an important role in various applications of natural language processing (NLP), such as machine translation (Hutchins, 1986; Su and Chang, 1990), speech recognition (Su, Chiang, and Lin, 1990; Seneff, 1992; Meteer and Gish,

1994), and information extraction (Hobbs et al., 1992; McDonald, 1992). It constructs the syntactic relationship of the words in an input sentence according to a given grammar which formally specifies the allowable syntactic structures in the language. In real applications, to correctly parse a sentence, a parser often encounters the problems resulted from the ambiguities in syntactic structure and the ill-formedness of the inputs.

Ambiguous syntactic structures are generated due to the implied ambiguity from language usage or due to the over-generation from the given grammar. The number of syntactic ambiguities of a sentence depends on the grammar. In practical applications, a sentence usually has thousands of ambiguities, and, on some occasions, the number of ambiguities may be greater than millions. To give a correct interpretation for the input sentence, a natural language parser must be able to choose the correct syntactic structure from such ambiguities. In the past, many algorithms have been proposed to resolve this problem and significant improvements have been observed (Briscoe and Carroll, 1993; Chiang, Lin, and Su, 1995).

The other problem in natural language parsing is the ill-formedness of inputs. An ill-formed sentence is the sentence that cannot be fitted into any well-formed syntactic structures generated by the grammar. The major sources of ill-formed inputs are (1) incorrect sentences resulted from typographical errors, OCR scanning etc, (2) unknown words which are not contained by the system dictionary and (3) the insufficient coverage of the grammar. Compared with the topic of syntactic disambiguation, the problem of ill-formed inputs is less investigated and is often ignored in experiment works. However, ill-formed inputs are inevitable in real applications because the incorrect sentences always exist in the real world and it is impossible to limit users using only the predefined artificial grammar and built-in vocabulary. Therefore, we focus on the problem of handling ill-formed sentences here.

In this paper, the *Phrase-Level-Building* (PLB) parsing mechanism is proposed to attack the ambiguity problem of ill-formed inputs by consulting the contextual information. In this framework, a parse tree is modeled as a set of *phrase-levels*. By decomposing a syntactic tree into phrase-levels, this mechanism regards the task of parsing a sentence as a task of building the phrase-levels for the sentence. A phrase-level here refers a set of terminals and nonterminals constructed at a particular snap shot of the parsing process. For example, the parser may construct a noun phrase "[N3 *Printer buffers*]" and

a verb phrase "[V2 *are made by DRAM*]" for the sentence "*Printer buffers are made by DRAM*". In this case, the phrase-level at this particular time consists of "[N3 *Printer buffers*]" and "[V2 *are made by DRAM*]". During parsing, a fast level-synchronous search mechanism is used to remove less likely phrase-levels. As a result, only the tree groups with large likelihood values are generated by the PLB parser. Whenever all active phrase-levels in the search beam cannot be further reduced by any grammar rules, the process of building phrase-levels is stopped and a probabilistic scoring function is used to select the best tree group. With this approach, the best tree group is selected within a wider scope (i.e., the whole sentence), and thus generates better result. Compared with the baseline system using the stochastic context-free grammar and the "leftmost longest phrase first" heuristics (which operates in a narrow scope), the proposed PLB approach improves the precision of brackets in the tree group from 69.37% to 79.49%. The recall of brackets is also improved from 78.73% to 81.39%.

2 Baseline System

In many frameworks (Jensen, Miller, and Ravin, 1983; Mellish, 1989; Seneff, 1992; Hobbs et al., 1992), the heuristics of preferring the longest phrase is used alone or with other system-dependent heuristics to further constraint the possible partial parses while parsing the ill-formed sentences. On the other hand, in some systems, natural language sentences are parsed by a left-corner parser (such as the LR parser), which uses the left context to limit the search space. If the input sentence is ungrammatical, only the partial parses beginning at the left are available in these systems. Thus, these systems usually parse the ill-formed input with the heuristics of selecting the leftmost longest phrase and then starting to parse from the subsequent word again. To make a comparative study, a baseline system is built to evaluate the performances of these two heuristic rules, *leftmost longest phrase first* (LLF) and *longest phrase first* (LF), on handling ill-formed sentences.

The baseline system consists of two components: a Cocke-Younger-Kasami (CYK) parser and a partial parse assembler. According to the Chomsky normal form (CNF) grammar (Chomsky, 1959), which is converted from a normal context-free grammar (CFG), the CYK parser efficiently parse the ill-formed sentence to all its possible partial parses; and every combination of the partial parses which covers all terminals will form a

tree group for the ill-formed sentence. Then, according to the adopted heuristic rule (LF or LLF), the partial parses are assembled by the partial parse assembler, in which the partial parses cover the same input words are ranked by the stochastic context-free grammar (SCFG) (Fujisaki et al., 1989; Ng and Tomita, 1991) with the probability parameters smoothed by the Good-Turing method (Good, 1953; Katz, 1987).

2.1 Evaluation Method

The performances of different approaches are measured by three factors: bracket precision, bracket recall and tree group accuracy. The parse tree in Figure 1 is used as an example to explain how to calculate the precision and recall for brackets. This parse tree has nine

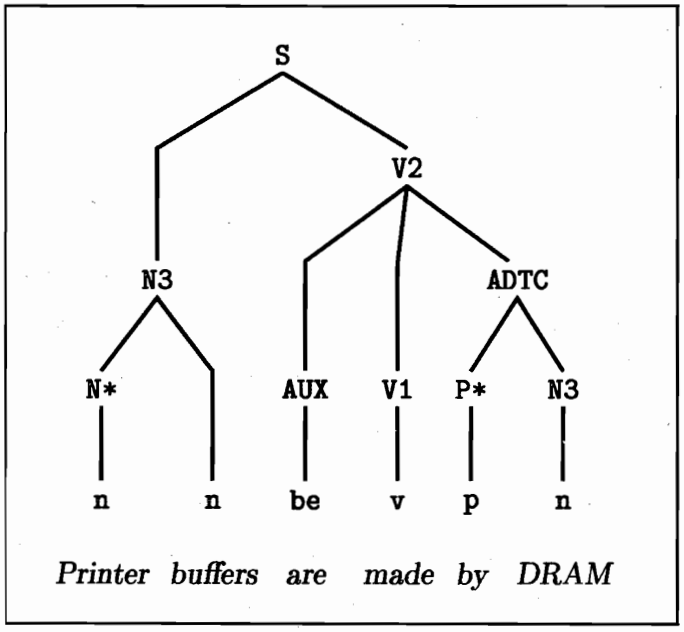


Figure 1: An example of the parse tree.

brackets shown below,

[N* *Printer*], [N3 *Printer buffers*], [AUX *are*],
[V1 *made*], [P* *by*], [N3 *DRAM*], [ADTC *by DRAM*],
[V2 *are made by DRAM*], [S *Printer buffers are made by DRAM*].

Each of the brackets corresponds to the application of a production (shown in a left-to-right depth-first traversal sequence).

The precision rate and the recall rate are computed as follows.

$$\text{bracket precision} = \frac{\text{number of exactly matched brackets}}{\text{number of brackets generated by parser}}$$

and

$$\text{bracket recall} = \frac{\text{number of exactly matched brackets}}{\text{number of brackets in correct parse trees}}.$$

It should be noticed that in many applications the grammar of a system to be measured may differ from the one used to parse the treebank (i.e. the database of correct parse tree). Therefore, the labels of the brackets are not taken into account in computing the precision or recall of brackets in most cases. However, in this task, both the system and the treebank use the same grammar. Thus, the labels of brackets are also taken into account while computing the precision and recall for brackets. In other words, we will regard two brackets as being "matched" only when they have the same label.

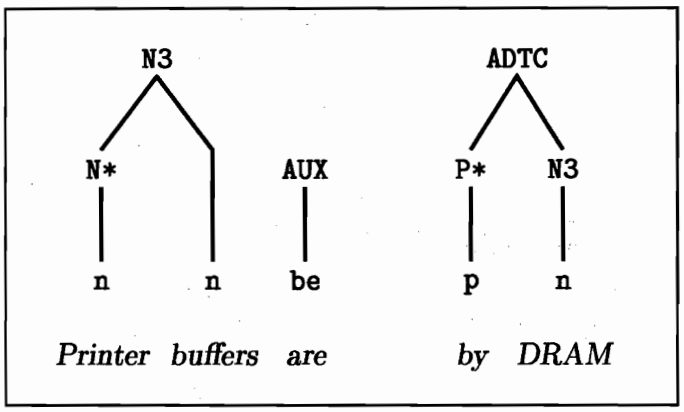


Figure 2: An example of the tree group.

Since the performance of a robust parser strongly depends on whether the parser can accurately partition the inputs or not, the factors "fragment precision" and "fragment recall" are also measured to reflect the performance of a robust parser on partitioning the ill-formed sentences. Here, fragments are the brackets which are not enclosed by any other brackets, i.e. the outmost brackets. For example, for the ill-formed sentence "*Printer buffers are by DRAM*" with the tree group in Figure 2, its three fragments are

[N3 *Printer buffers*], [AUX *are*], [ADTC *by DRAM*].

On the other hand, since an ill-formed sentence cannot be parsed to a full parse tree, the conventional parse tree accuracy is replaced with the tree group accuracy, which is computed as

$$\text{tree group accuracy} = \frac{\text{number of exactly matched tree groups}}{\text{number of sentences}},$$

where "exactly matched tree groups" means that all the tree groups consist of the same partial parses.

2.2 Simulation Results and Discussions

In the baseline system, 8,727 well-formed sentences, collected from computer manuals, and their correct parse trees are used as the training data. The average length of these sentences is about 13 words. All the training sentences are parsed by a context-free grammar provided by the Behavior Design Corporation. This grammar consists of 29 terminals, 140 nonterminals and 1,013 production rules. To test the performance of the baseline system, 200 ill-formed sentences and their tree groups are used as the testing data. The average length of the testing sentences is about 13 words.

	Bracket and its label		Tree group accuracy (%)	Parsing time (sec./sent.)
	Precision (%)	Recall (%)		
LF	67.98	77.54	16.5	2.16
LLF	69.37	78.73	16.5	2.16

Table 1: The performances of the baseline system with "longest phrase first" (LF) and "leftmost longest phrase first" (LLF) heuristics.

Table 1 lists the simulation results with different heuristic rules. The precision and recall of the labeled brackets are given in the first and second columns. The third column shows the accuracy rate of tree group and the last column gives the average processing time for parsing a sentence with a "SUN SPARC station ELC". The experiment results show that LLF slightly outperforms LF. This is because the LF, compared with the LLF heuristics, is more likely to grab the words belonging to the neighboring phrases. In fact, due to preferring a larger partial parse than a smaller one, the LF heuristics always

	Number of fragments		Precision (%)	Recall (%)
	Total	Matched		
Treebank	605	—	—	—
LF	331	160	48.3	26.5
LLF	355	179	50.4	29.6

Table 2: The performances of LLF and LF on fragments.

partitions a sentence into as few fragments as possible. As shown in Table 2, the number of fragments generated by using the LLF heuristics is only 355, which is much smaller than that of the correct fragments. But, the number of fragments generated by using the LF heuristics is even smaller. In other words, assembling partial parses with the LF heuristics produces more inadequate partitions than with the LLF heuristics.

In fact, both the LLF and LF heuristic rules use rather coarse knowledge to assemble partial parses. They always append the largest partial parses, either the leftmost one or a global one, to the tree group, regardless of the context of the partial parse. Therefore, the performance is not really satisfactory. In the next section, a Phrase-Level-Building parsing algorithm is proposed to parse the ill-formed inputs by consulting more contextual information.

3 PLB Parsing

In this section, a *Phrase-Level-Building* (PLB) parsing algorithm is proposed to parse an ill-formed sentence using contextual information in wider scope. This algorithm treats the parsing process as the procedure of building a set of *phrase-levels*. During parsing, a fast level-synchronous search mechanism is used to cut down the search space. Instead of using heuristics, the final parse trees are ranked by a probabilistic scoring function which makes use of the contextual information in the phrase-levels. The details of this algorithm is described in the following sections.

3.1 Phrase-Levels of a Parse Tree

The basic idea of PLB parsing is to model a syntactic tree as a set of phrase-levels. Figure 3 is an example to show the relations between a syntactic tree and its phrase-levels. As shown in this figure, the syntactic tree for the sentence *"Printer buffers are made by DRAM"* is decomposed into six phrase-levels. The lowest one, $\mathbf{L}_1$, corresponds to the input

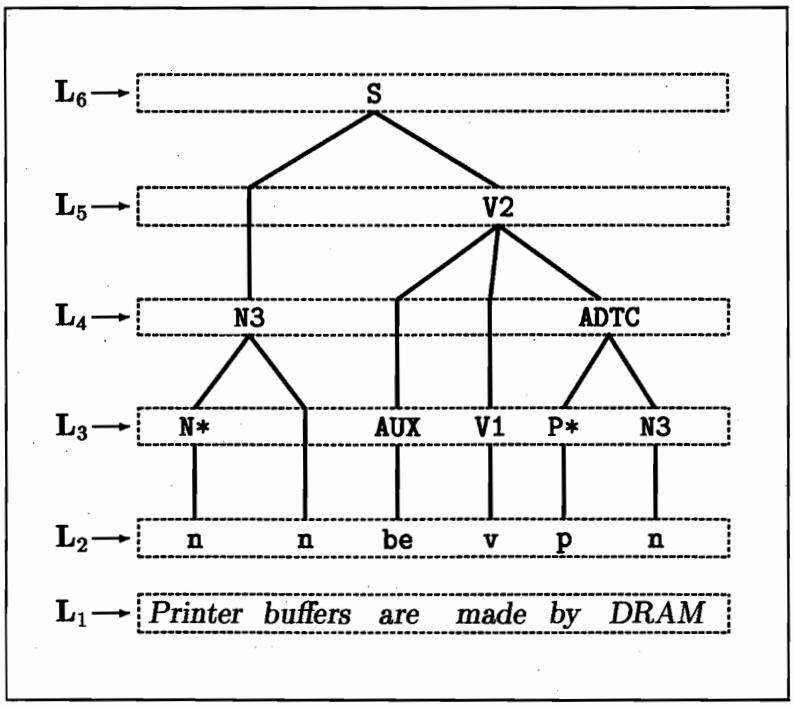


Figure 3: A syntactic tree and its phrase-levels.

words. The second phrase-level consists of the parts-of-speech of the input words. The other phrase-levels are sequences of grammar symbols (i.e. terminals and nonterminals) which are obtained by applying some grammar rules on the grammar symbols of the previous phrase-level. As a result, the parsing process can be considered as the procedure of building the phrase-levels from $\mathbf{L}_1$ to $\mathbf{L}_6$ in a bottom-up manner.

Every phrase-level in a parse tree is obtained by applying some production rules on the phrase-level immediately preceding the current phrase-level. In Figure 3 $\mathbf{L}_3$ has six grammar symbols and is denoted as $\mathbf{L}_3$ = { `N*` `n` `AUX` `V1` `P*` `N3` }. By applying the productions "`N3` → `N*` `n`" and "`ADTC` → `P*` `N3`" on the leftmost two and the rightmost two grammar symbols respectively, $\mathbf{L}_3$ is built up to $\mathbf{L}_4$, which has four grammar symbols

and is denoted as $\mathbf{L}_4 = \{$ `N3 AUX V1 ADTC` $\}$. As a result, the parse tree in Figure 3 can be represented as $\mathbf{T} = \{\mathbf{L}_1, \mathbf{R}_1, \mathbf{L}_2, \mathbf{R}_2, \cdots, \mathbf{L}_5, \mathbf{R}_5, \mathbf{L}_6\}$, where $\mathbf{R}_i$ denotes a sequence of actions which are applied to build $\mathbf{L}_i$ up to $\mathbf{L}_{i+1}$. For example, to build $\mathbf{L}_4$ from $\mathbf{L}_3$ in Figure 3, $\mathbf{R}_3$ contains two actions, which corresponds to applying the productions "`N3` $\rightarrow$ `N* n`" and "`ADTC` $\rightarrow$ `P* N3`" on the leftmost two and the rightmost two grammar symbols in $\mathbf{L}_3$. The detailed definition of action will be described in the following section.

3.2 Scoring a Parse Tree

The likelihood of the parse tree of N phrase-levels can be derived as follows.

$$\begin{aligned} P(\mathbf{T} = \mathbf{L}_1, \mathbf{R}_1, \cdots, \mathbf{L}_{N-1}, \mathbf{R}_{N-1}, \mathbf{L}_N | w_1^n) &= \prod_{i=2}^{N} P(\mathbf{L}_i, \mathbf{R}_{i-1} | \mathbf{L}_{i-1}, \mathbf{R}_{i-2}, \cdots, \mathbf{L}_1) \\ \approx \prod_{i=2}^{N} P(\mathbf{L}_i, \mathbf{R}_{i-1} | \mathbf{L}_{i-1}) &= \frac{P(\mathbf{L}_N)}{P(\mathbf{L}_1)} \prod_{i=2}^{N} P(\mathbf{R}_{i-1}, \mathbf{L}_{i-1} | \mathbf{L}_i) . \end{aligned} \tag{1}$$

In the above equation, the prior probability $P(\mathbf{L}_N)$ is introduced by applying the Bayesian formula because $P(\mathbf{L}_N)$ is regarded as useful information to assemble partial parses to a tree group. For example, the likelihood of the tree group in Figure 2 is considered related to the likelihood of a nonterminal sequence "`N3 AUX ADTC`".

Note that $P(\mathbf{R}_{i-1}, \mathbf{L}_{i-1} | \mathbf{L}_i) = P(\mathbf{R}_{i-1} | \mathbf{L}_i)$ because $\mathbf{L}_{i-1}$ is uniquely determined by $\mathbf{R}_{i-1}$ and $\mathbf{L}_i$. Therefore, the Equation 1 can be written as

$$P(\mathbf{T} | w_1^n) \approx \frac{P(\mathbf{L}_N)}{P(\mathbf{L}_1)} \prod_{i=2}^{N} P(\mathbf{R}_{i-1} | \mathbf{L}_i) . \tag{2}$$

Since $P(\mathbf{L}_1)$ is the prior probability of the input sentence, it is the same for all competing parse trees and can be ignored without changing the ranking order of the likelihood values of the competing parse trees. Suppose there are n symbols $\{A_1, \cdots, A_n\}$ in the phrase-level $\mathbf{L}_N$, the probability $P(\mathbf{L}_N)$ can be approximated by a trigram model as follows.

$$P(\mathbf{L}_N = A_1, \cdots, A_n) = \prod_{A_j \in \mathbf{L}_N} P(A_j | A_1, \cdots, A_{j-1}) \approx \prod_{A_j \in \mathbf{L}_N} P(A_j | A_{j-2}, A_{j-1}) \tag{3}$$

The probability term $P(\mathbf{R}_{i-1}|\mathbf{L}_i)$ in Equation (2) accounts for the actions which are applied to build $\mathbf{L}_i$ from $\mathbf{L}_{i-1}$. Alternatively, it could be regarded as the probability of applying the rewriting rules in $\mathbf{R}_{i-1}$ at the phrase-level $\mathbf{L}_i$ from a top-down point of view. Before deriving $P(\mathbf{R}_{i-1}|\mathbf{L}_i)$, the notations for the actions of $\mathbf{R}_{i-1}$ will be defined first. Let $\{\rho_1, \cdots, \rho_m\}$ denote the m actions of $\mathbf{R}_{i-1}$. The j-th action ρ_j will be denoted as $\rho_j = \langle r; t \rangle$, where the rule argument r denotes the rule applied by ρ_j; and the position argument t is the index of the reduced symbol in $\mathbf{L}_i$. For example, to build $\mathbf{L}_4$ from $\mathbf{L}_3$ in Figure 3, the production rules " `N3` $\rightarrow$ `N* n` " and " `ADTC` $\rightarrow$ `P* N3` " are applied respectively. In this case, the corresponding actions are $\rho_1 = \langle r = \texttt{N3} \rightarrow \texttt{N* n}; t = 1 \rangle$ and $\rho_2 = \langle r = \texttt{ADTC} \rightarrow \texttt{P* N3}; t = 4 \rangle$. The reduced symbols are `N3` and `ADTC`, which are the 1st and 4th symbols in $\mathbf{L}_4$ respectively. Therefore, the position arguments t of these two actions are 1 and 4 respectively. Figure 4 gives a more clear illustration for the relationship of those phrase-levels and actions.

$\mathbf{L}_4$	= { `N3`	`AUX V1`	`ADTC`	}	ρ_1= ⟨ `N3` → `N* n` ; 1 ⟩
$\mathbf{R}_3$	= { ρ_1		ρ_2	}	ρ_2= ⟨ `ADTC` → `P* N3` ; 4 ⟩
$\mathbf{L}_3$	= { `N* n`	`AUX V1`	`P* N3`	}	

Figure 4: Two phrase-levels and their corresponding actions.

With these notations, the conditional probability $P(\mathbf{R}_{i-1}|\mathbf{L}_i)$ can be derived as follows. Let $A_1^n \equiv \{A_1, \cdots, A_n\}$ denote the n symbols in $\mathbf{L}_i$ and $\rho_1^m \equiv \{\rho_1, \cdots, \rho_m\}$ denote the m actions in $\mathbf{R}_{i-1}$. Then, the conditional probability $P(\mathbf{R}_{i-1}|\mathbf{L}_i)$ can be approximated as

$$P(\mathbf{R}_{i-1}|\mathbf{L}_i) = P(\rho_1^m|A_1^n) = \prod_{j=1}^{m} P\left(\rho_j|\rho_1^{j-1}, A_1^n\right) \approx \prod_{\rho=\langle r;t\rangle \in \mathbf{R}_{i-1}} P\left(r|A_{t-1}^{t+1}\right), \quad (4)$$

where we assume that the action $\rho_j = \langle r; t \rangle$ depends on the its local context $A_{t-1}, \cdots, A_{t+1}$.

According to Equations (2)-(4), the likelihood of a parse tree is approximated as follows.

$$P(\mathbf{T}|w_1^n) \approx \frac{1}{P(\mathbf{L}_1)} \times \prod_{A_j \in \mathbf{L}_N} P(A_j|A_{j-2}, A_{j-1}) \times \prod_{i=1}^{N-1} \prod_{\rho=\langle r;t\rangle \in \mathbf{R}_i} P\left(r|A_{i+1,t-1}^{i+1,t+1}\right). \quad (5)$$

Note that, in the above equation, the notation $A_{i+1,t-1}^{i+1,t+1}$ denotes the sequence "$A_{i+1,t-1}$ $A_{i+1,t}$ $A_{i+1,t+1}$", which represents the $(t-1)$-th symbol, the t-th symbol and the $(t+1)$-th symbol in $\mathbf{L}_{i+1}$ respectively. These three symbols are the local context of an action $\rho = \langle r; t \rangle$ in $\mathbf{R}_i$. As mentioned before, the probability $P(\mathbf{L}_1)$ can be ignored while ranking the likelihoods of parse trees because it is the same for all competing parse trees. Therefore, the parse tree scoring function $S_{\mathrm{PT}}(\cdot)$ is defined as follows.

$$S_{\mathrm{PT}}(\mathbf{T}|w_1^n) \equiv \prod_{A_j \in \mathbf{L}_N} P(A_j|A_{j-2}, A_{j-1}) \times \prod_{i=1}^{N-1} \prod_{\rho=\langle r;t\rangle \in \mathbf{R}_i} P\left(r|A_{i+1,t-1}^{i+1,t+1}\right) \qquad (6)$$

The parameters in this scoring function are smoothed by the Good-Turing smoothing method.

3.3 The PLB Parsing Mechanism

The PLB parser parses an input sentence as building the phrase-levels for the sentence. The building process is illustrated in Figure 5, where we assume there are only two ambiguities for every phrase-level candidate while expanding it up to a higher level. As shown in Figure 5, up to the 4th phrase-level, there are eight different *partial trees* (i.e. tree groups), each of which consists of four phrase-levels and is represented by one of the eight paths. Since the number of paths increases exponentially, it is infeasible to exhaustively travel all possible paths during parsing. Thus, the beam search is adopted to find the most likely paths.

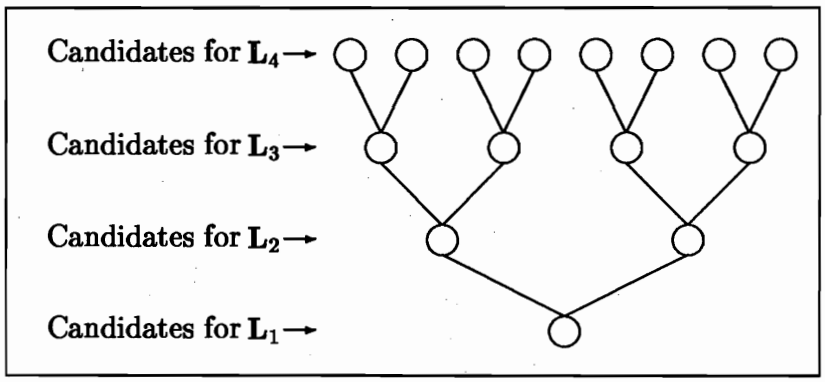


Figure 5: An illustration of building the phrase-levels up.

To efficiently carry out the beam search, we require a scoring function which can rank the candidates of every phrase-level in very short time. However, the scoring function in Equation (6) will spend too much computation time in ranking the candidates of a phrase-level because all possible candidates of the phrase-level must be expanded and scored. Thus, a time-saving scoring function is proposed in this section to rapidly find the potential candidates of a phrase-level, and Equation (6) will be used only after the final level is reached. In other words, two different scoring functions are used during the parsing process and during the final best tree selection process, respectively.

To make the derivation of the scoring function more clear, another representation form of parse trees is introduced in the following. From another point of view, the process of building a phrase-level $\mathbf{L}_i$ up to a higher phrase-level $\mathbf{L}_{i+1}$ can be considered as segmenting $\mathbf{L}_i$ into segments and then transforming these segments into $\mathbf{L}_{i+1}$. For example, as shown in Figure 3, building $\mathbf{L}_3$ = { N* n AUX V1 P* N3 } up to $\mathbf{L}_4$ = { N3 AUX V1 ADTC } is equivalent to segmenting $\mathbf{L}_3$ to four segments as { [N* n] [AUX] [V1] [P* N3] } and then transforming these four segments to $\mathbf{L}_4$ = { N3 AUX V1 ADTC }. Figure 6 gives an illustration of such segmentation and transformation, where the notation $\mathbf{C}_3$ denotes the segmented phrase-level obtain by segmenting $\mathbf{L}_3$. Therefore, during parsing, a partial tree of i phrase-levels can be represented by a sequence of unsegmented and segmented phrase-levels as $\{\mathbf{L}_1, \mathbf{C}_1, \mathbf{L}_2, \mathbf{C}_2, \cdots, \mathbf{L}_{i-1}, \mathbf{C}_{i-1}, \mathbf{L}_i\}$.

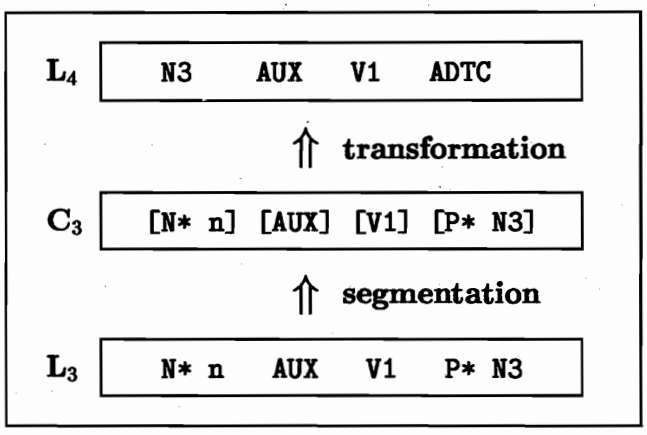


Figure 6: Parsing by segmentation and transformation.

Based on the above representation, the likelihood of a partial tree of i phrase-levels is computed as

$$
\begin{aligned}
& P\left(\mathbf{L}_1, \mathbf{C}_1, \cdots, \mathbf{L}_{i-1}, \mathbf{C}_{i-1}, \mathbf{L}_i | w_1^n\right) \\
&= \prod_{j=2}^{i} P\left(\mathbf{L}_j, \mathbf{C}_{j-1} | \mathbf{L}_{j-1}, \mathbf{C}_{j-2}, \cdots, \mathbf{L}_1\right) \approx \prod_{j=2}^{i} P\left(\mathbf{L}_j, \mathbf{C}_{j-1} | \mathbf{L}_{j-1}\right).
\end{aligned} \tag{7}
$$

The approximation in the above equation is based on the assumption that the segmentation and transformation results (i.e. $\mathbf{L}_j$ and $\mathbf{C}_{j-1}$) only depend on the previous phrase-level (i.e. $\mathbf{L}_{j-1}$). According to the above equation, the scoring function $S_{\text{BS}}(\cdot)$ is defined as follows to evaluate the score of a partial tree of i-th phrase-levels, where the subscript BS in $S_{\text{BS}}(\cdot)$ denotes "beam search".

$$
S_{\text{BS}}\left(\mathbf{L}_1, \mathbf{C}_1, \cdots, \mathbf{L}_{i-1}, \mathbf{C}_{i-1}, \mathbf{L}_i\right) \equiv S_{\text{lex}}\left(\mathbf{L}_2, \mathbf{C}_1 | \mathbf{L}_1\right) \times \prod_{j=3}^{i} S_{\text{syn}}\left(\mathbf{L}_j, \mathbf{C}_{j-1} | \mathbf{L}_{j-1}\right), \tag{8}
$$

where $S_{\text{lex}}\left(\mathbf{L}_2, \mathbf{C}_1 | \mathbf{L}_1\right) = P\left(\mathbf{L}_2, \mathbf{C}_1 | \mathbf{L}_1\right)$ denotes the lexical score of the part-of-speech sequence of $\mathbf{L}_2$; $S_{\text{syn}}\left(\mathbf{L}_j, \mathbf{C}_{j-1} | \mathbf{L}_{j-1}\right) = P\left(\mathbf{L}_j, \mathbf{C}_{j-1} | \mathbf{L}_{j-1}\right)$ denotes the syntactic score corresponding to the j-th phrase-level $\mathbf{L}_j$. The lexical and syntactic scores are provided by the lexical and syntactic modules respectively. The following sections give the details of these two modules.

3.3.1 Lexical Module

The lexical module is basically a statistical tagger (Church, 1989) which finds the most likely part-of-speech sequence for the input sentence. The likelihood of a part-of-speech sequence $\mathbf{L}_2$ for the input word sequence $\mathbf{L}_1$ is computed according to the widely-used trigram model (Church, 1989; Lin, Chiang, and Su, 1995) as follows.

$$
\begin{aligned}
P\left(\mathbf{L}_2, \mathbf{C}_1 | \mathbf{L}_1\right) &= P\left(c_1^n | w_1^n\right) = P\left(w_1^n | c_1^n\right) \frac{P\left(c_1^n\right)}{P\left(w_1^n\right)} \\
&\approx \frac{1}{P\left(w_1^n\right)} \prod_{j=1}^{n} P\left(w_j | c_j\right) P\left(c_j | c_{j-2}, c_{j-1}\right),
\end{aligned} \tag{9}
$$

where n is the number of words in the input sentence, w_j is the j-th input word and c_j denotes the part-of-speech for the j-th input word. Since the probability $P\left(w_1^n\right)$ is a constant, it can be ignored without changing the ranking order of the likelihood probabilities

of those competing part-of-speech sequences. Therefore, the scoring function $S_{\text{lex}}(\cdot)$ for the lexical module is defined as

$$S_{\text{lex}}(\mathbf{L}_2, \mathbf{C}_2|\mathbf{L}_1) \equiv \prod_{j=1}^{n} \{P(w_j|c_j) P(c_j|c_{j-2}, c_{j-1})\}, \tag{10}$$

where w_j is the j-th input words in $\mathbf{L}_1$ and c_j is the j-th part-of-speech in $\mathbf{L}_2$.

3.3.2 Syntactic Module

The syntactic module is responsible for ranking the phrase-level candidates which are one level higher than the given phrase-level. The likelihood of a phrase-level candidate $\mathbf{L}_i$ for the given phrase-level $\mathbf{L}_{i-1}$ is computed as follows.

$$P(\mathbf{L}_i, \mathbf{C}_{i-1}|\mathbf{L}_{i-1}) = P(\mathbf{L}_i|\mathbf{C}_{i-1}, \mathbf{L}_{i-1}) P(\mathbf{C}_{i-1}|\mathbf{L}_{i-1}) = P(\mathbf{L}_i|\mathbf{C}_{i-1}) P(\mathbf{C}_{i-1}|\mathbf{L}_{i-1}). \tag{11}$$

Let $A_1, \cdots, A_n$ be the n symbols in $\mathbf{L}_{i-1}$ and $\alpha_1, \cdots, \alpha_m$ be the m segments in $\mathbf{C}_{i-1}$. Then, the first probability term on the right-hand side of Equation (11) is approximated as

$$\begin{aligned} P(\mathbf{C}_{i-1}|\mathbf{L}_{i-1}) &= P(\alpha_1^m|A_1^n) = \prod_{j=1}^{m} P\left(\alpha_j|\alpha_1^{j-1}, A_1^n\right) \approx \prod_{j=1}^{m} P(\alpha_j|\alpha_{j-2}, \alpha_{j-1}) \\ &\approx \prod_{j=1}^{m} P(\alpha_j|\Gamma_{\text{R2}}(\alpha_{j-2}\alpha_{j-1})), \end{aligned} \tag{12}$$

where $\Gamma_{\text{R2}}(\alpha_{j-2}\alpha_{j-1})$ denotes the rightmost two symbols of $\alpha_{j-2}\alpha_{j-1}$.

The last probability term on the right-hand side of Equation (11) is derived as

$$\begin{aligned} P(\mathbf{L}_i|\mathbf{C}_{i-1}) &= P(A_1^m|\alpha_1^m) = \prod_{j=1}^{m} P\left(A_j|A_1^{j-1}, \alpha_1^m\right) \approx \prod_{j=1}^{m} P(A_j|\alpha_{j-1}, \alpha_j, \alpha_{j+1}) \\ &\approx \prod_{j=1}^{m} P(A_j|\Gamma_{\text{R1}}(\alpha_{j-1}), \alpha_j, \Gamma_{\text{L1}}(\alpha_{j+1})), \end{aligned} \tag{13}$$

where $\Gamma_{\text{R1}}(x)$ and $\Gamma_{\text{L1}}(x)$ denote the the rightmost symbol and the leftmost symbol in x respectively.

According to Equations (11)-(13), the syntactic scoring function $S_{\text{syn}}(\mathbf{L}_i, \mathbf{C}_{i-1}|\mathbf{L}_{i-1})$ is defined as follows.

$$S_{\text{syn}}(\mathbf{L}_i, \mathbf{C}_{i-1}|\mathbf{L}_{i-1}) \equiv \prod_{j=1}^{m} P(\alpha_j|\Gamma_{\text{R2}}(\alpha_{j-2}\alpha_{j-1}))\, P(A_j|\Gamma_{\text{R1}}(\alpha_{j-1}), \alpha_j, \Gamma_{\text{L1}}(\alpha_{j+1})), \quad (14)$$

where α_j is the j-th segment in $\mathbf{C}_{i-1}$ and A_j is the j-th symbol in $\mathbf{L}_i$. The parameters used in Equation (10) (the lexical scoring function) and Equation (14) (the syntactic scoring function) are smoothed by the Good-Turing smoothing method. Using this scoring function, the syntactic module can rapidly rank the possible candidates for a given phrase-level.

3.4 Simulation Results and Discussions

The PLB parsing mechanism uses the scoring function $S_{\text{BS}}(\cdot)$, Equation (8), to rapidly rank the candidates of phrase-levels and remove the less likely ones during constructing the tree groups. Therefore, only the tree groups with high probability are generated. Then, the scoring function $S_{\text{PT}}(\cdot)$, Equation (6), is used to select the best one from the generated tree groups. The performances of the PLB parsing in the testing set with various beam widths are listed in Table 3. In general, the accuracy rates and the parsing time increases while the beam width increases. The accuracy rates almost saturate after the beam width exceeds 20. On the other hand, the parsing time rapidly increases when the beam width is greater than 20. Therefore, the beam width of 20 is recommended for the PLB parsing in this task.

The results of the baseline system with LLF heuristics (selecting the leftmost longest phrase first) are also listed in Table 3 for comparison. It is obvious that the PLB approach significantly outperforms the baseline system. Even using a very small beam width, the PLB approach achieves better results than the baseline system in terms of the precision and recall of brackets as well as on the accuracy rate of the whole tree group. Since the search space is cut down via a probabilistic scoring function during parsing, the PLB approach can rapidly select the most possible combination of the partial parses for ill-formed inputs. Therefore, the PLB approach with a small search beam width can parse the inputs faster than the baseline system. However, current experiments still

cannot claim that the PLB approach is more time-saving than other systems with LLF heuristics, because the LLF heuristics can be implemented by left-corner parsers which are theoretically more efficient than the CYK parser used in the baseline system. But, since the PLB approach can obtain better results than the LLF heuristics within one second, the LLF heuristics is no more attractive even if it could be implemented by a faster parser.

	Beam width	Bracket and its label		Tree group accuracy (%)	Parsing time (sec./sent.)
		Precision (%)	Recall (%)		
PLB	3	78.52	79.27	24.5	0.46
	5	78.22	80.56	25.5	0.66
	10	78.78	80.74	26.0	1.17
	20	79.49	81.39	27.5	2.51
	50	80.08	80.92	26.5	9.75
	100	80.11	80.98	27.0	35.93
LLF		69.37	78.73	16.5	2.16

Table 3: The performances of PLB parsing in the testing set with various beam widths.

Table 3 shows that the improvement on bracket precision rate achieved by the PLB approach is better than the improvement on bracket recall rate. For instance, compared to the LLF heuristics, the PLB approach with beam width of 20 improves the bracket precision rate by 10.12% (from 69.37% to 79.49%); while it only improves the bracket recall rate by 2.66% (from 78.73% to 81.39%). To further explore the reason, more detailed data are given in Table 4. It indicates that there are 3,343 brackets in the hand-parsed treebank (i.e. the testing set of the 200 ill-formed sentences). The second row shows that there are 3,794 brackets in the parse trees assembled by the LLF heuristics. However, among these 3,794 brackets, only 2,632 brackets (i.e. 69.37%) are correct. Such a low precision rate results from the fact that the heuristics of selecting the longest phrase, either the leftmost one or the global one, usually selects undesirable partial parses. On the contrary, the PLB approach assembles the partial parses according to the statistical information and, consequently, selects the desirable configuration in more cases.

Selecting the longest phrase also causes the baseline system to inadequately partition

	Number of brackets		Precision (%)	Recall (%)
	Total	Matched		
Treebank	3,343	—	—	—
LLF	3,794	2,632	69.37	78.73
PLB	3,423	2,721	79.49	81.39

Table 4: The detailed results of the baseline system and the PLB approach.

	Number of fragments		Precision (%)	Recall (%)
	Total	Matched		
Treebank	605	—	—	—
LLF	355	179	50.4	29.6
PLB	656	350	53.4	57.9

Table 5: The performances of LLF and PLB on fragments.

the ill-formed sentences. As shown in Table 5, there are 605 fragments in the 200 ill-formed sentences. However, the baseline system partitions these ill-formed sentences into only 355 fragments, which is much smaller than the number of that they should be. This is due to the fact that, while assembling partial parses, the baseline system does not consider the contextual information. It always prefers a larger partial parse than a smaller one, and consequently partitions a sentence into as few fragments as possible. Thus, the baseline system has a very low recall rate for fragments. On the other hand, due to the use of statistical contextual information, the PLB approach can more accurately partition the ill-formed sentences. It partitions the 200 ill-formed sentences into almost the same number of fragments as that they should be. Besides, the number of matched fragments generated by the PLB approach is much larger than that generated by the baseline system. Therefore, the PLB approach has a significantly higher recall rate for fragments than the baseline system (57.9% v.s. 29.6%).

In summary, by using the statistical contextual information, the proposed PLB approach outperforms the baseline system to a great extent. With the beam width of 20, the PLB approach significantly improves the precision of brackets in the tree group from 69.37% to 79.49%. The recall of brackets is also improved from 78.73% to 81.39%.

4 Conclusions

Parsing the ill-formed input usually suffers from the ambiguity problem more deeply than parsing the grammatical sentences. The ambiguities of an ill-formed sentence include all possible tree groups, each of which is a combination of the partial parses jointly generating the input sentence. Since the number of possible tree groups is very large, it is infeasible to do disambiguation by enumerating all of them. In the past, the heuristics of preferring a larger phrase is used (or with other heuristics) to limit the number of partial parses. However, this heuristic rule, although simple to implement, fails to achieve satisfactory performance because the longest phrase is not always the correct phrase.

This paper presents a Phrase-Level-Building (PLB) parsing mechanism to resolve the ambiguity problem of ill-formed inputs. In this framework, a parse tree is modeled as a set of phrase-levels for being explored in a wider scope. By decomposing a syntactic tree into phrase-levels, this mechanism regards the task of parsing a sentence as a task of building the phrase-levels from the sentence. During parsing, a level-synchronous scoring function is used to remove less likely phrase-levels. As a result, instead of enumerating all possible tree groups, the PLB parser only generates the more likely ones. Whenever all active phrase-levels in the search beam cannot be further reduced by the grammar rules, the process of building phrase-levels is stopped and a probabilistic scoring function is used to select the best tree group. Compared with the baseline system using stochastic context-free grammar and the "leftmost longest phrase first" heuristics, the proposed PLB approach improves the precision rate of brackets in the tree group from 69.37% to 79.49%. The recall rate of brackets is also improved from 78.73% to 81.39%.

Acknowledgement

This paper is a partial result of the project No. 3P11200 conducted by ITRI under sponsorship of the Ministry of Economic Affairs, R.O.C.

References

Briscoe, Ted and John Carroll. 1993. Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, 19(1):25–59.

Chiang, Tung-Hui, Yi-Chung Lin, and Keh-Yih Su. 1995. Robust learning, smoothing, and parameter tying on syntactic ambiguity resolution. *Computational Linguistics*, 21(3).

Chomsky, Noam. 1959. On certain formal properties of grammars. *Information and Control*, 2:137–167.

Church, Kenneth Ward. 1989. A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of ICASSP*, pages 695–698, Glasgow, May 23-26.

Fujisaki, T., F. Jelinek, J. Cocke, E. Black, and T. Nishino. 1989. A probabilistic parsing method for sentence disambiguation. In *Proceedings of the International Workshop on Parsing Technologies*, pages 85–94, Pittsburgh, Pennsylvania, USA, 28–31 Aug.

Good, I. J. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264.

Hobbs, Jerry R., Douglas E. Appelt, John Bear, and Mabry Tyson. 1992. Robust processing of real-world natural-language texts. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 186–192, Trento, Italy, 31 Mar. – 3 Apr.

Hutchins, W. J. 1986. *Machine Translation: Past, Present, Future.* West Sussex, England: Ellis Horwood Limited.

Jensen, K., G. E. Heidorn L. A. Miller, and Y. Ravin. 1983. Parse fitting and prose fixing: Getting a hold on ill-formedness. *American Journal of Computational Linguistics*, 9(3–4):147–160, July–December.

Katz, S. M. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. Acoustic, Speech, Signal Processing*, ASSP-34(3):400–401, March.

Lin, Y.-C., T.-H. Chiang, and K.-Y. Su. 1995. The effects of learning, parameter tying and model refinement for improving probabilistic tagging. *Computer Speech and Language*, 9:37–61.

McDonald, David D. 1992. An efficient chart-based algorithm for partial-parsing of unrestricted texts. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 193–200, Trento, Italy, 31 Mar. – 3 Apr.

Mellish, Chris S. 1989. Some chart-based techniques for parsing ill-formed input. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, pages 102–109, Vancouver, British Columbia, Canada, 26–29 June.

Meteer, Marie and Herbert Gish. 1994. Integrating symbolic and statistical approaches in speech and natural language applications. In *Proceedings of the Workshop on The Balancing Act Combining Symbolic and Statistical Approaches to Language*, pages 69–75, Las Cruces, New Mexico, USA, 1 July.

Ng, See-Kiong and Masaru Tomita. 1991. Probabilistic LR parsing for general context-free grammars. In *Proceedings of the Second International Workshop on Parsing Technologies*, pages 154–163, Cancun, Mexico, 13–15 Feb.

Seneff, Stephanie. 1992. Robust parsing for spoken language system. In *Proceedings of the 1992 International Conference on Acoustics, Speech and Signal Processing*, pages 189–192, San Fransisco, California, USA, 23–26 Mar.

Su, K.-Y., T.-H. Chiang, and Y.-C. Lin. 1990. A unified probabilistic score function for integrating speech and language information in spoken language processing. In *Proceedings of the 1990 International Conference on Acoustics, Speech and Signal Processing*, pages 901–904, Kobe, Japan, Nov. 19-22.

Su, Keh-Yih and Jing-Shin Chang. 1990. Some key issues in designing MT systems. *Machine Translation*, 5(4):265–300.

THE APPLICATION OF THE SIMILARITIES BETWEEN THE MORPHEMES OF THE ENGLISH AND CHINESE LANGUAGES TO REPRESENT CHINESE CHARACTERS PHONETICALLY WITH ENGLISH LETTERS TO FACILITATE COMPUTER APPLICATIONS MANUALLY AND BY VOICE WITH THE CHRACTER-BASED LANGUAGES CHINESE, JAPANESE AND KOREAN

Dr. Stanley K. Chan
Research Institute for Computer Application of East Asian Languages
7798 Starling Drive, Suite 207, San Diego, CA 92123, USA
Telephone: (619) 571-5285 Fax (619) 571-5992
E-mail stanleykchan@msn.com

ABSTRACT

This paper presents a new methodology for entering, manually or by voice, Chinese characters into a computer and processing them with the same ease as English words by utilizing the similarities between the morphemes of the Chinese and English languages to represent Chinese characters and radicals phonetically and unambiguously with English letters or local phonetic symbols such as Zhuyin Zimu, Hiragana and Hangul.

1. Problems With The Use Of Chinese Characters In Computer Applications

Unlike English, the written form of the Chinese, Japanese and Korean languages contains unique square-shaped symbols, called Chinese characters (actually Hanzi for Chinese, Kanji for Japanese, Hanja for Korean pronunciation. "Chinese" will be used throughout this article for clearer presentation.), that present significant obstacles to their use in computer applications. Firstly, input of such characters is daunting as it is impractical to make a keyboard with the thousands of keys that would be needed to represent the Chinese character set. Secondly, numerous instances exist in which many characters share an identical pronunciation.

Historically, the limiting factor in developing a satisfactory system for input of Chinese characters using English alphabet letters has been the existence of multiple groups of large numbers of homonymous Chinese characters. The input of a string of English letters frequently fails to provide an unambiguous reference to the desired Chinese character, generating incorrect Chinese characters for some strings of English letters.

Inadequacies in existing programs that attempt to solve this problem make them less efficient than writing Chinese characters by hand. They display all the homonymous characters, ten at a time, upon entry of a given set of letters, requiring that the typist scroll through the homonymous characters to search for the desired character. This time-consuming process is tiring to the eye and prone to errors since the typist may have to search as many as 250 homonymous characters to make a single correct selection.

2. Multiple Groups Of Numerous Homonyms

The average number of homonyms per pronunciation is 30 if we use 13,000 as a typical number of commonly-used Chinese characters. Theoretically, this problem should be reduced if intonation is taken into consideration, reducing the number of homonyms per pronunciation to 7.5. In practice, however, it is a different story. This is due to the fact that the Chinese people seem to favor pronunciations ji, qi, xi, yi, wu, yu, zhi, chi, shi, zi, ci, si, and a few others. These preferences generate over 100 groups of 40 or more homonyms[1]. This phenomenon requires a search process for the desired character when inadequate data (typically the pronunciation or pronunciation/ intonation combination of a character) is used as input.

The most effective way to address this problem, for the time being, seems to be the charactword input method discussed later in this paper.

[1] Please see Appendix 1. Not all groups with 40 or more homonyms are included. Groups with fewer than 40 are not included at all. Zhuyin Zimu with * indicates that the appropriate Zhuyin Zimu is not available in the database due to the fact that Mainland China has incorporated it into the one used in the chart.

3. Surprising Similarities Between Chinese Characters And English Words

Chinese characters can be seen to contain analogs to the prefixes and suffixes of English words. Although these components of Chinese characters are called different names and are treated differently in the rules of pronunciation, these "prefixes" and "suffixes" have been an integral part of written East Asian languages for thousands of years. Scholars call these Chinese character components "radicals," meaning "parts of a Chinese character." Radicals are systematically taught in Japan and Taiwan alongside Chinese characters and phonetic signs, namely Katakana and Hirogana in Japan and Zhuyin Zimu in Taiwan[2].

Generally, radicals fall into two categories. In one category, the radical resembles a shape or bears a meaning related to the character of which it is a part. When such a radical is related to shapes, it is called the "pictographic radical," for example, "木" , (pronounced "mu," in Chinese, meaning "wood" or "tree.") When a radical is related to meaning, it is called an "ideographic radical." We shall refer to radicals of these types as "P/I" radicals. Radicals in the second category denote the approximate or exact pronunciation of characters. These are phonetic radicals such as "每 "(pronounced "mei" in Chinese). The combination of a P/I radical and a phonetic radical makes a character. For example, combining the aforementioned radicals creates the character 梅 , meaning "plum." This combining of radicals to make a character is analogous to creating an English word by combining a prefix and suffix.

4. Application Of "Charactwords" For Manual And Voice Input Of Chinese Characters

If we use English letters to spell out the P/I radicals according to their pronunciations, using them as *prefixes*, and do the same with the phonetic radicals using them as *suffixes*, we can form "words" for the Chinese, Japanese and Korean languages. For example, combining the P/I radical 木 (spelled "mu") and the phonetic radical 每 (spelled "mei") gives us the "word" for character 梅 , spelled "mu-mei." This provides a simple methodology for using these created "words" for computer input of Chinese characters. We shall refer to these newly-created words as

[2] The following sets of phonetic signs are used by the Chinese, Japanese and Koreans respectively: Zhuyin Zimu, Hiragana or Katakana, and Hangul. Only Mainland China officially uses a Latin style alphabet to aid teaching the pronunciation of Chinese characters.

"charactwords" whose function it is to represent Chinese characters for easy computer input and processing. For voice input, all relevant parts of a charactword should be pronounced to provide all the data necessary to unambiguously represent the desired character. This method can be implemented cost-effectively as it does not require a large database, complex programming or strict noise control in the immediate working area.

5. Charactwords Are Actually Chinese Characters Expressed In The Form Of Alphabetic Letters Or Phonetic Signs

Regardless of which written phonetic signs are used to spell out Chinese characters, whether Hangul, Kana, Zhuyin Zimu, or a Latin-based alphabet, the charactwords so created remain logically and theoretically Chinese characters, with P/I radicals represented by the prefixes and phonetic radicals represented by the suffixes. The failure of existing programs to provide a satisfactory solution to the input of Chinese characters arises because such programs use insufficient data to describe the characters. These programs incorporate only one-half of the CHINESE character data available, building their databases solely upon pronunciation (and, sometimes the intonation) data while ignoring the equally important data related to the meaning of the radical and the shape of the character. By contrast, using a charactword database which contains all the relevant character data[3] yields an efficient solution. Further, with such a database, programming a computer to display the correct Chinese character upon input of the appropriate charactword becomes a simple task.

6. Markers And Intonation Indicators Further Differentiate The Charactwords From One Another

Chart 1 illustrates how a system of markers and intonation indicators differentiates the spelling of characters sharing the same P/I radical and pronunciation. Markers are input by simply repeating the last key of the charactword spelling as many times as necessary (although seldom more than four) to uniquely identify or represent the desired Chinese character. In a sense, the markers give each charactword a distinct look. Additionally, intonation indicators

3. Please refer to Chart 1 on the next page for the *complete* version of the charactword for Chinese characters used in Chinese, Japanese and Korean languages.

required for the Chinese language are incorporated in the form of the first four Chinese Mandarin consonants[4] appended to the spelling of a charactword, representing the first, second, third, and fourth intonation, respectively[5]. Together with the prefixes and suffixes, the markers and the intonation indicators can be used to create not only unique charactwords for all characters, but also charactwords for all the different ways a character can be pronounced. Please refer to subsection 9 "A Charactword Can Be Tailored For All Different Ways In Which A Chinese Character Is Pronounced" for details.

Chart 1

How Markers and Intonation Indicators Help To Unambiguously Differentiate Complex Homonyms

Row number	The country where the characters are homony-mous	Homony-mous Chinese charact-ers	Corres-pond-ing charact-word	Intonation indicators (bold letters) added to both prefix and suffix of the charactwords	Markers (italic letters) appended to the charact-words to unambiguously differentiate them from one another
1	China	試	yan-shi	yanp-shif	yanp-shif ㄧㄢˊㄕˋ
2	China	誓	yan-shi	yanp-shif	yanp-shif*f* ㄧㄢˊㄕˋㄈ
3	China	諶	yan-shi	yanp-shif	yanp-shif*ff* ㄧㄢˊㄕˋㄈㄈ
4	China	諡	yan-shi	yanp-shif	yanp-shif*fff* ㄧㄢˊㄕˋㄈㄈㄈ
5	China	謚	yan-shi	yanp-shif	yanp-shif*ffff* ㄧㄢˊㄕˋㄈㄈㄈㄈ
6	China	識	yan-shi	yanp-shif	yanp-shif*fffff* ㄧㄢˊㄕˋㄈㄈㄈㄈㄈ
7	Japan	杠	ki-ko	not applicable	ki-ko き* こ
8	Japan	杞	ki-ko	not applicable	ki-ko*o* き* こ こ

[4] In Mandarin Chinese (the official dialect of Mainland China, Taiwan and Singapore,) no pronunciation of a Chinese character ends with one of the first four consonants, so the presence of one of these consonants at the end of the spelling of a charactword can be readily recognized as an indicator of intonation.

[5] The indicator for the first intonation is traditionally omitted, therefore only the second, third and fourth consonants are used for the corresponding intonations.

9	Japan	杭	ki-ko	not applicable	ki-k*ooo* き* こ こ こ
10	Japan	枯	ki-ko	not applicable	ki-k*oooo* き* こ こ こ こ
11	Korea	杞	muk-ki	not applicable	muk-ki 목*기
12	Korea	枝	muk-ki	not applicable	muk-k*i* 목*기
13	Korea	棋	muk-ki	not applicable	muk-k*ii* 목*기
14	Korea	機	muk-ki	not applicable	muk-k*iii* 목*기

7. Explanation Of Chart 1

Rows 1 - 6 contain six Chinese characters with an identical P/I radical, pronunciation and intonation. Column 6 shows how markers, indicated in italics, unambiguously differentiate all six charactwords. Similar results are shown for Japanese characters in rows 7 -10, and for Korean in rows 11 - 14. National or local phonetic symbols are hand written alongside the Latin-based alphabetic letters in column 6.

The rule for assigning markers is based upon the number of pen strokes comprising a character: the character with the least pen strokes receives no marker, and the number of markers increments one at a time in relation to the number of additional pen strokes comprising a character, with the most markers assigned to the character made up of the most pen strokes. When two or more characters have the same number of pen strokes, the order of appearance for the Chinese language is based upon their order of appearance in Kang Xi Zi Dian (Emperor Kang Xi's Dictionary, 康熙字典), which is the case for the characters in rows 3 and 4. Similarly, The Modern Reader's Japanese-English Dictionary (最新漢英辭典) is used for Japanese, and for Korean we have selected 新活 用玉篇that does not have an English title. The number of markers becomes a non-issue with our careful software design. Please refer to subsection 11 for details.

8. Input Of The Prefix Is Efficient

Many of the 214 P/I radicals can be input with one or two keystrokes. For example, 木 will

appear on the screen by typing just ㄇ (the full spelling is ㄇㄨ) in Zhuyin, or "m" (the fullspelling is "mu") in the Pinyin system. Voice recognition capability could further simplify charactword input. An advantage of the charactword system is that most of the P/I radicals are commonly-used Chinese characters, and all have associated pronunciations. Only 38 of the 214 P/I radicals are not phonetically recognized by a typical high school graduate in Taiwan[6]. In practical computer input applications, any necessary reference can be provided by a large card at the computer terminal that displays all 214 radicals and highlights the 38 uncommon radicals.

9. A Charactword Can Be Tailored For All Different Ways In Which A Chinese Character Is Pronounced

There are occasions when a character can be pronounced in more than one way. An example is the Chinese character, 好, which is ordinarily pronounced "hao*m*[7]," meaning "good." When it is pronounced as "hao*p*[8]" preceding an adjective, it means "very." It can also be pronounced "hao*f*[9]", meaning "fond of" or "hobby, "depending on the context. When "nyu" (or "nu", the pronunciation of the P/I radical, 女, for 好,) and "haom", "nyu" and "haop" or "nyu" and "haof" are entered, 好 will be displayed. This is because in our database, either pronunciation (hence the charactword) is linked to the same character.

The same technique can be applied to the Japanese language where some Kanji characters can be pronounced in as many as 9 different ways. For an example, 鸚can be pronounced as "ryu, ru, bo, hyo, mu, kyo, gu, ryo and hibari." By collecting all different charactwords representing all the different pronunciation in our database, character, 鸚, will be displayed when one of the following sets is typed into the computer equipped with our software: "tori[10]" and "ryu"; "tori" and "ru"; "tori" and "bo"; "tori" and "hyo"; "tori" and "mu"; "tori" and "kyo"; "tori" and "gu"; "tori" and "ryo"; "tori" and "hibari".

6 According to our random survey of the Taiwanese students studying in the San Diego area.

7 Letter "*m*" is the indicator for the third intonation. Please refer to footnote 5 in subsection 6 for details.

8 Letter "*p*" is the indicator of the second intonation. Please refer to footnote 5 in subsection 6 for details.

9 Letter "*f*" is the indicator for the fourth intonation. Please refer to footnote 5 in subsection 6 for details.

10 "Tori" is the Japanese pronunciation of P/I radical 鳥, the radical of character 鸚。

10. Charactwords Inherit All The Data Of Chinese Characters

As discussed earlier, there are four kinds of data in most Chinese characters: 1. pronunciation; 2. intonation; 3. meaning; and 4. shape or construction.

It is easy to see that a charactword might contain the first three of the four types of data, but it is a little bit difficult to see how a string of alphabetic letters can denote the "shape" of a character, until we run into Complex Homonyms (CH).

Complex Homonyms are characters that share the identical P/I radicals, pronunciation and intonation, such as 誓試諡 諟 諡識 . As all of us can see, the clever ancient Chinese used the "shape difference" to unambiguously differentiate each character from the others. We can do the same with a charactword by adding marker(s) to it, one at a time, making each and every charactword look different from one another, much as English words "knight" and "night".

11. The Wonders Of Software Eliminate The Guess Work: No Need To Depend On Memory Or Dictionaries

This leads to a couple of questions: how can I remember how many markers there are in the charactword for character "諟"? Do I have to refer to my Chinese dictionary all the time? The answer? You do not need to memorize the number of markers for any charactword at all, nor do you need to use your dictionary. That is because through our software design, all members of a group of Complex Homonyms can be displayed on the screen once the charactword for a member of the group is entered. In this extreme example of 誓試諡諟 諡識 , once "yanp" and "shi" are typed in the environment of the software, ***all*** of these characters are displayed. Suppose you want to use "諟", after typing "yanp" and "shi", all you need to do is to repeatedly type "i" three more times until the highlight moves onto it. (This does not take much time since your right middle finger is already on that key). Then hit the space bar, and, "諟" will be in your text. All Complex Homonyms will be handled this way in our software. Although there are groups of CHs, there are not as many members in each group as shown in this

extreme example, especially with just the commonly-used characters. Leave your dictionary on the shelf when you use our software. You don't need it.

12. Typing Of The Charactwords Can Be Simplified To Empower Non-Career Typists To Achieve The Typing Speed Of A Professional

At first glance, the numerous "markers" in the charactwords seem to require a lot of typing. Closer examination reveals this is not so. A marker is created by the repetition of the last key of the charactword, so the typist does not need to move his/ her finger- he/ she can just keep pressing the last key, moves the highlight to a different character, once per keystroke. The examples in Chart 1 are rare and extreme. They are intended to demonstrate the usefulness of markers in creating unique charactwords to unambiguously represent each one of a group of complex homonymous characters, which is otherwise very difficult to accomplish. Our experience shows that more than 90% of the characters in Chinese can be unambiguously differentiated with just the regular charactwords. Here are some examples: each of the characters 基(ji),欺(qi),稀(xi),醫(yi)[11] has over 140 homonyms[12], but by first typing their P/I radicals, 土(tu),欠(qian),禾(he), 酉(you)[13], respectively,then the Pinyin of the characters, you will have precisely these four characters on the screen, eliminating more than 140 homonyms for each of them. The time it takes to type the markers, the intonation indicators and even prefixes will become a non-issue when users become well acquainted with the charactwords and move to the next stage of inputting described below.

With careful software design and a broad database, the input of charactwords can be greatly simplified. For an example, most of the Chinese idioms (成語) containing four or more characters can be input with an average of just 1 to 1.25 keystrokes for each character, using the acronyms of their corresponding suffixes. For the speech parts such as nouns, pronouns, verbs, adjectives and adverbs consisting of two characters, a modified but theoretically sound acronym

11 The English letters in the parenthesis are actually the Pinyin letters denoting the pronunciation of these characters.

12 The actual numbers are over 220, 140, 180 and 250 respectively. Please refer to Appendix 1 for details.

13 The English letters in the parenthesis are actually the Pinyin letters denoting the pronunciation of these P/I radicals. Work has begun to reduce the number of keys required for each P/I radicals. It looks promising that the number of keys for the most commonly used P/I radicals, such as 人，心，手，金，木，水，火，土，日，月，石，艸 etc. can be reduced to just one keystroke.

system of their suffixes and prefixes can be used to input them at an average of 4.5 to 5 keystrokes per character. These can also be applied to voice input. Whether inputting manually or by voice, an average East Asian can enter more than 1,000 characters per hour into a computer, a satisfactory speed for most of us[14]. When people become accustomed to the charactwords, the typing speed will increase to about 3,000 characters per hour, rivaling the speed of today's career typists[15].

13. Typing Speed Of Non-career Typists Can Be Calculated

I wish to express my gratitude to the members of ROCLING X Program Committee for bringing to my attention that the contention that the average Chinese can type 3,000 characters per hour is controversial. Because it is common knowledge that non-career typists can write Chinese characters much faster than they can type, further analysis is called for to accurately model the typing speed ranges. Our latest investigation revealed that an experienced[16] American typist can type 45 words per minute. An average English word consists of 6[17] alphabetic letters. This translates into 270 keystrokes per minute.

A charactword consists of an average of 4.5 to 5 Zhuyin Zimu[18]. Assuming an average "experienced" Chinese can type as fast as his/her American counterpart, it will mean that he/ she can type about 54 Chinese characters per minute. The hourly speed would, therefore, be 3,240 characters.

All skills improve with time and practice. Typing is no exception. The speed of "typing" Chinese characters via charactword-based input method will increase as time goes by because all impeding elements are eliminated.

Since our prototype uses Zhuyin Zimu, we are currently unable to establish the speed for typing the Pinyin alphabets. We shall conduct a test to determine the speed for typing Pinyin

[14] An average person is one who is semi-familiar with keyboard layout and can type 1.75 keys per second. At an average rate of 5 keystrokes per character, more than 1,260 characters can be entered into a computer per hour by an average person.

[15] Most of the people can type an average 4.5 to 6 keys per second after using the keyboard consistently for about 3 to 6 months. At that speed, one can enter more than 3,600 characters into a computer per hour.

[16] "Experienced" means being familiar with the keyboard layout *and* having 480 hours of practice in typing.

[17] Most experts say that the average letters in an English word is 7. I use 6 in order to be conservative.

[18] This takes into account of the abbreviation of the P/I radicals and two-character vocabularies, but not the four-character idioms.

once our database for Pinyin is complete. It is, however, expected to be somewhat slower than typing Zhuyin Zimu (when typing individual characters,) because some of the latter frequently requires fewer keystrokes to represent characters. The typing speed will be the same for both Zhuyin and Pinyin when inputting vocabularies of two or more characters.

14. The Advantages Of Our Software

. The most important aspect of our software is the theory upon which it is based. As previously mentioned, our software is based on charactwords that contain all relevant data needed to describe Chinese characters including the various ways to pronounce them. This means that the pronunciation, intonation, meaning, and even the shape of any given Chinese character is taken into consideration in the making of a charactword. A Chinese character is unique because of its appearance, only one of the four data used to create a charactword. This is why a charactword can unambiguously represent the character or its various pronunciations.

This completeness of data in the charactwords makes our software faster in producing the desired character and easier to use by the general public. It is faster, because it eliminates searching for the desired character among scores or even hundreds of homonyms. It is easier to use for two reasons. One, the charactwords resemble the characters in three ways: the pronunciation, the intonation and the meaning. Two, charactwords are more logically organized, with the meaningful part as the prefixes and the phonetic part as the suffixes. By using markers, the charactwords can even have unique shapes. This built-in familiarity to the East Asian public is especially true in Taiwan because the P/I radicals are systematically taught beginning in elementary school.

As discussed earlier, our software also has a very complete database that includes special, yet logical and easy-to-understand coding of two, three, four or multiple-character vocabularies, that can be input with far fewer keystrokes than many individual characters. It also incorporates coding of various ways a character can be pronounced as discussed in subsection 9.

The most popular Chinese software in the US is Twinbridge, according to my research. Twinbridge is a versatile software product that provides many ways of entering a Chinese

character. The one that compares closest to ours is Zhuyin. In entering individual characters, it takes an average of 22.5 seconds to produce a commonly-used character with an "untrained[19]" version of the software. It took only 2.1 seconds with our "prototype" software operated by an "inexperienced" typist. Twinbridge fared better with two-character vocabularies input than single character input, but ours can do better[20]. With idioms of four or more characters, ours can generate a character with less than one second per character[21] theoretically.

Appendix 2 summarizes the comparison of these two software products that can be categorized into two separate classes, differentiated by their underlying methodologies and the degree to which they make use of all available character description data. Please note that the typing speed with our software seem to be the same, 2.1 seconds for all three categories, they are actually different: the first category is for a *single* character; the second *doubles* the number of characters; the third *quadruples* the number of characters. The reason for this "coincidence"? The number of keystrokes for all three categories are about the same: four to five.

15. A Vehicle And An Opportunity For The East Asians

For the vast majority of the East Asians who choose not to use memory-intensive input methods, the charactword-based input system can be a viable alternative, because it is easy to learn and use without depending on memory. It will also make computers easier to use, and more *useful*, to the residents of this region. All in all, the charactword-based input method could be an ideal vehicle for the 1.5 billion Chinese, Japanese and Korean people to access the information highway alongside the Americans and Europeans, providing East Asia an opportunity to make another powerful contribution to humanity again.

[19] This product has a mechanism that moves the typed Chinese characters to the front of their homonyms. So it is easier to find the Chinese characters after using the product for a while. At that point, the product is "trained."

[20] While our database for this feature is not yet complete, we known that one set of this type of vocabularies can be entered by typing just four or five keys.

[21] The database for this type of input is not complete at the moment, but we know that over 90% of the four-character idioms can be inputted with just four keys. The rest of these type of idioms can be entered with just five or six keys, but it will be more likely five than six keys. It really does not matter whether it will be five or six keys, because the keys after the 4th one will be the repetition of the 4th key.

Appendix 1 Homonym groups and the approximate number of homonyms belonging to each group

Pronunciation	Number of homonyms	Pronunciation	Number of homonyms	Pronunciation	Number of homonyms	Pronunciation	Number of homonyms
bi ㄅㄧ	120	lian ㄌㄧㄢ	50	jing ㄐㄧㄥ	60	chen ㄔㄣ	50
bao ㄅㄠ	40	lin ㄌㄧㄣ	40	ju ㄐㄩ	120	cheng ㄔㄥ	50
bei ㄅㄟ	40	ling ㄌㄧㄥ	60	jun ㄐㄩㄣ	40	chu ㄔㄨ	50
biao ㄅㄧㄠ	40	lei ㄌㄟ	50	qi ㄑㄧ	140	shi ㄕ	120
bo ㄅㄛ	80	gan ㄍㄢ	40	qian ㄑㄧㄢ	90	shan ㄕㄢ	50
pi ㄆㄧ	70	ge ㄍㄜ	60	qiang ㄑㄧㄤ	40	shen ㄕㄣ	50
pu ㄆㄨ	40	gu ㄍㄨ	70	qiao ㄑㄧㄠ	50	shu ㄕㄨ	40
di ㄉㄧ	80	gui ㄍㄨㄟ	50	qiu ㄑㄧㄡ	60	zi ㄗ	70
du ㄉㄨ	40	ke ㄎㄜ	50	qu ㄑㄩ	70	ci ㄘ	40
ta ㄊㄚ	40	kui ㄎㄨㄟ	50	quan ㄑㄧㄢ	40	si ㄙ	60
ti ㄊㄧ	50	he ㄏㄜ	60	xi ㄒ	180	suo ㄙㄨㄛ	40
tu ㄊㄨ	40	han ㄏㄢ	60	xian ㄒㄧㄢ	110	sui ㄙㄨㄟ	40
tong ㄊㄨㄥ	50	hao ㄏㄠ	40	xiang ㄒㄧㄤ	40	yan ㄧㄢ	150
tang ㄊㄤ	40	hu ㄏㄨ	90	xiao ㄒㄧㄠ	60	yang ㄧㄤ	50
mao ㄇㄠ	40	hui ㄏㄨㄟ	40	xing ㄒㄧㄥ	40	yao ㄧㄠ	80
mi ㄇㄧ	50	huo ㄏㄨㄛ	40	xu ㄒㄩ	80	you ㄧㄡ	80
mei ㄇㄟ	40	huan ㄏㄨㄢ	60	xun ㄒㄩㄣ	60	yu ㄩ	200
mo ㄇㄛ	60	huang ㄏㄨㄤ	50	zhi ㄓ	160	yue ㄩㄝ*	40
fan ㄈㄢ	50	hung ㄏㄨㄥ	60	zhan ㄓㄢ	50	yuan ㄩㄢ	70
fen ㄈㄣ	50	ji ㄐㄧ	220	zhen ㄓㄣ	70	yun ㄩㄣ	50
fei ㄈㄟ	50	jia ㄐㄧㄚ	70	zhou ㄓㄡ	40	wu ㄨ	120
feng ㄈㄥ	40	jian ㄐㄧㄢ	120	zhu ㄓㄨ	100	wei ㄨㄟ	130
fu ㄈㄨ	170	jiao ㄐㄧㄠ	80	chi ㄔ	90	wan ㄨㄢ	50
ni ㄋㄧ	50	jie ㄐㄧㄝ*	100	chan ㄔㄢ	60	e ㄜ	70
li ㄌㄧ	160	jin ㄐㄧㄣ	60	chou ㄔㄡ	40	ao ㄠ	40
liao ㄌㄧㄠ	40					an ㄢ	40

Appendix 2 Comparison of Twinbridge And Our Software

Name	Single character input			2-character vocabulary input			4-character idiom input			Fonts
Name of software	Input method	Is the result unique and what is the implication?	Average time to generate a character	Input method	Is the result unique and what is the implication?	Average time to generate a vocabulary	Input method	Is the result unique and what is the implication?	Average time to generate an idiom	Stability of fonts and the reason why
Twinbridge	Pronunciation of the character	No. Searching for the desired character is required	22.5 seconds	Continuous typing of: 1. the pronunciation of the two characters, but not necessarily all of the alphabets, or 2. acronyms of the pronunciation	No. Searching for the desired character is required	12.8 seconds	Typing the acronyms of the pronunciation	Not enough data to determine. It is likely to need to search if two or more idioms share the same acronym	Unable to determine due to insufficent data	Poor, because it operates in the English Windows where the fonts are single-byte while the Chinese fonts are double-byte.
Asian Language Software Solution (Ours)	Character-word based input	Yes. You get exactly the character you desire.	2.1 seconds or less (Zhuyin Zimu input only)	Typing of the acronym and marker(s)	Yes. You get exactly the vocabulary you desire.	2.1 seconds or less (Good for both Zhuyin and Pinyin)	Typing of the acronym and marker(s)	Yes. You get exactly the idiom you desire.	2.1 seconds less (Good for both Zhuyin and Pinyin)	Excellent font stability because it operates in the Chinese Windows.

A Multivariate Gaussian Mixture Model for Automatic Compound Word Extraction

[+]Jing-Shin Chang and [+*]Keh-Yih Su

[+]Department of Electrical Engineering, National Tsing-Hua University, Hsinchu, Taiwan.

*Behavior Design Corporation, 2F, No. 5, Industrial East Road IV,

Science-Based Industrial Park, Hsinchu, Taiwan.

[+]shin@hermes.ee.nthu.edu.tw, [+*]kysu@bdc.com.tw

Abstract

An improved statistical model is proposed in this paper for extracting compound words from a text corpus. Traditional terminology extraction methods rely heavily on simple filtering-and-thresholding methods, which are unable to minimize the error counts objectively. Therefore, a method for minimizing the error counts is very desirable. In this paper, an improved statistical model is developed to integrate parts of speech information as well as other frequently used word association metrics to jointly optimize the extraction tasks. The features are modelled with a multivariate Gaussian mixture for handling the inter-feature correlations properly. With a training (resp. testing) corpus of 20715 (resp. 2301) sentences, the *weighted precision & recall* (WPR) can achieve about 84% for bigram compounds, and 86% for trigram compounds. The F-measure performances are about 82% for bigrams and 84% for trigrams.

1. Compound Word Extraction Problems

1.1 Motivation

Compound words are very common in technical manuals. Including such technical terms in the system dictionary beforehand normally improves the performance of an NLP system significantly. In a machine translation system, for instance, the translation quality will be greatly improved if such unknown compounds are identified and included before the translation process begins. On the other hand, if a compound is not in the dictionary, it might be translated incorrectly [Chen 88]. For example, the Chinese translation of 'green house' is not the composite of the Chinese translations of 'green' and 'house'. Furthermore, the number of parsing *ambiguities* will also increase due to the large number of possible parts of speech combinations for the individual words if such new compounds are unregistered. It will then reduce the *accuracy* rate in disambiguation, degrade the processing or translation *quality* and increase the *processing time*.

In addition, for some NLP tasks, such as machine translation, a computer-translated manual is usually concurrently processed by several posteditors in practical operations. Therefore, maintaining the consistency of the translated terminologies among different post-editors is very important. If all the terminologies can be entered into the dictionary beforehand, the consistency can be automatically maintained, the translation quality can be greatly improved, and lots of post-editing time and *consistency* maintenance cost can be saved.

Since compounds are rather productive and new compounds are created from day to day, it is impossible to exhaustively store all compounds in a dictionary. Furthermore, identifying the compounds by human inspection is too costly and time-consuming for a large input text. Therefore, spotting and updating such

terminologies before translation without much human effort is important; an *automatic* and quantitative tool for extracting compounds from the text is thus seriously required.

1.2 Technical Problems in Previous Works

The extraction problem can be modeled as a two-class classification problem, in which potential compound candidates are classified into either the compound class or the non-compound class. Many English or Chinese extraction issues had been addressed in the literature [Church 90, Calzolari 90, Bourigault 92, Wu 93, Smadja 93, Su 94b, Tung 94, Chang 95, Wang 95, Smadja 96]. Our focus will be on statistical methods for English compound word extraction, since statistical approaches have many advantages for large-scale systems in automatic training, domain adaptation, systematic improvement, and low maintenance cost.

Most statistical approaches [Church 90, Smadja 93, Tung 94, Wang 95, Smadja 96] for terminology extraction rely on word association metrics, such as frequency [Wang 95, Smadja 96], mutual information [Church 90], dice metrics [Smadja 93] and entropy [Tung 94] to identify whether a group of words is a potential compound (or highly associated collocate). The mechanisms for applying such features are often based on simple filtering-and-thresholding statistical tests; a compound candidate will be filtered out (or classified as non-compound) if its association metric is below a threshold; when multiple features are available, the features are usually applied one-by-one independently with different heuristically determined thresholds. Such approaches can be implemented easily, and encouraging results were reported in various works. However, there are several technical problems with such filtering approaches.

First of all, most simple word association features, such as frequency and mutual information, can only indicate whether an n-gram (i.e., a group of n words) is highly associated; however, high association does not always implies that it is a compound, since there are other syntactic (and even semantic) constraints which will also produce highly associated n-grams. For instance, the word pair "is a" has sufficiently high frequency of occurrence and high mutual information. Nevertheless, it is not a compound word since such a construct is produced due to syntactic reasons. Many long collocates extractable by such filtering methods are also of this category [Smadja 96]. Therefore, many highly associated non-compound n-grams might be mis-recognized as compounds.

Although it is known that *syntactic information* is useful in resolving such problems, there are few works for integrating high level syntactic or semantic features, such as parts of speech, with known word association metrics in a *simple* and effective way. A part of speech related metric is therefore proposed in this paper to formulate the syntactic constraints among the constituents of potential compound candidates. Such integration between word association metrics and syntactic constraints in a uniform formulation is important, since syntactic constraints are closely related to the generation of the compounds, and it is desirable to apply simple statistical tests based on such features, instead of using complicated syntactic processing.

Second, since the association features are often applied independently for filtering even with multiple features available, it is impossible to jointly use all discrimination information to acquire the best system performance. For instance, by filtering out low frequency candidates and then filtering out candidates with low mutual information, we may filter out low frequency candidates which actually have high mutual information. If the filtering mechanism is based on *both* frequency and mutual information, the system performance is expected to be better. In fact, it is well known that the performance is usually improved if multiple features are jointly considered, instead of using a single feature or applying multiple features

independently. Therefore, what is really important is an automatic approach which could combine all available features for acquiring the best performance in the extraction task.

However, several factors must be carefully considered in order to enjoy the discrimination information provided by multiple association features. For instance, many features proposed in the literatures are highly correlated. Therefore, the correlations among the association features must be included into the statistical model in order to acquire the best achievable performance. In this work, we will therefore use (a mixture of) multivariate Gaussian density functions to incorporate the effects of the inter-feature correlation. Furthermore, it is desirable to use only the most discriminative features and reject features that are either non-discriminative or redundant with respect to other more discriminative features when combining the features. In this paper we therefore propose an integrated method, which select the most appropriate features automatically, for combining a set of useful features. In particular, optimization based on frequency, mutual information, dice metric, contextual entropy and parts of speech information will be surveyed.

To sum up, current terminology extraction researches do not fully exploit techniques for (1) integrating high level syntactic information in a simple and effective way, (2) combining useful features jointly for discrimination. To attack such problems, the parts of speech information, which encodes syntactic constraints, is integrated with several known word association metrics in one unified scoring mechanism. The correlations among the features are taken into consideration in designing the classifier. A feature selection mechanism is used for incorporating as many discriminative and non-redundant features as possible so that the terminology extraction task is based on the joint observations of the most discriminative features. A minimum error classifier, based on likelihood ratio test, is used as the basis for minimizing the classification error in the extraction task.

In the following sections, we will therefore focus on the general issues to design a good minimum error classifier, which jointly considers a set of association features for achieving minimum classification error. The simulation result shows that the proposed approach gives promising results. The tool is also observed to be useful in cooperating with a machine translation system [Chen 91].

2. Optimal Classifier Design

2.1 Optimization Criteria in Compound Extraction

In a compound retrieval task, it is desirable to recover from the corpus as many real candidates as possible; in addition, the extracted compound word list should contains as little 'false alarm' (i.e., incorrect candidates) as possible. The ability to extract real candidates in the corpus is defined in terms of the recall rate, which is the percentage of real compounds that are extracted to the compound list by the classifier; on the other hand, the ability to exclude false alarm from the extracted compound list is defined in terms of the precision rate, which is the percentage of real compounds in the extracted compound list. Let $n_{\alpha\beta}$ be the number of class-α input tokens which are classified as class-β (α, β = 1 for compound, and 2 for non-compound, respectively), and, let n_1 represent the number of real compounds in the corpus. The precision p and recall r are defined as follows:

$$p = \frac{n_{11}}{n_{11} + n_{21}}$$

$$r = \frac{n_{11}}{n_{11} + n_{12}} = \frac{n_{11}}{n_1}$$

The precision and recall rates are, in many cases, two contradictory performance indices especially for simple filtering approaches. When one of the performance index is raised, another index might degrade. To make fair comparison in performance, a joint performance indice or criterion function O(p,r) of the precision (p) and recall (r) rates is usually used to evaluate the system performance, instead of evaluating precision or recall alone. In the following sections, the *weighted precision & recall* (WPR) and the *F-measure* (FM) will be adopted as the optimization criteria. The weighted precision and recall (WPR), which reflects the average of these two indices, is proposed here as the weighting sum of the precision and recall rates:

$$WPR(w_p{:}w_r) = w_p * p + w_r * r \qquad (w_p + w_r = 1)$$

where w_p, w_r are weighting factors for precision and recall, respectively. The F-measure (FM) [Appelt 93, Hirschman 95, Hobbs 96], defined as follows, is another joint performance metric which allows lexicographers to weight precision and recall differently:

$$FM(\beta) = \frac{(\beta^2+1)pr}{\beta^2 p + r}$$

where β encodes user preference on precision or recall. When β is close to 0 (i.e., FM is close to p), the lexicographer prefers the system with higher precision; when β is large, the lexicographer prefers the system with higher recall. We will use Wp=Wr=0.5 and β=1, throughout this work, which means that no particular preference over precision or recall is imposed. If β=1, FM reduces to $\frac{2pr}{p+r}$, which appreciates the balance between precision and recall in the sense that equal precision and recall is most preferred if $p + r$ is identical. With the optimization criteria defined, our goal is to design an optimal classifier which could maximize the WPR and FM.

2.2 Task Definition for Optimal Classifier Design

Conventional extraction methods tend to use a list of word association related constraints for filtering out candidates of low likelihood based on certain word association metrics and empirical thresholds for the metrics. Unfortunately, there are no simple rules, other than trial-and-error, for such methods to acquire the optimal thresholds for acquiring the required precision or recall performance. In general, when the precision is raised by using high thresholds the recall degrades, and *vice versa*. The lexicographers could only use such tools by guessing. It is very difficult to automatically fit the lexicographers' preference on the precision-vs-recall performance. Such difficulty can be resolved if we can design an optimal classifier for automatically maximizing the performance criterion, such as WPR or FM, which encode user preference in the pre-specified weights.

The extraction problem can be regarded as a two-class classification problem in which each n-gram candidate is assigned either the compound label or the non-compound label based on the feature vector **x** associated with the candidate. To design a compound extractor is therefore equivalent to designing a discrimination function g(**x**;Λ) (which is capable of scoring how likely a candidate comes from the compound class), and using a set of decision rules to decide which n-gram candidate is a compound. (The symbol Λ refers to the parameters of the discrimination function, such as distributional means or variances of the probability density functions used in a statistical model.)

Different discrimination functions and decision rules will classify the input candidates differently, and

thus have different performance in terms of a performance criterion. Designing an optimal classifier for a particular criterion function is therefore equivalent to finding a partition of the feature space into the decision regions for the compound class and non-compound class; feature vectors belonging to the compound decision regions are classified as compound, otherwise, they are classified as non-compound. Our main task is therefore to design an optimal classifier (or equivalently the corresponding discrimination function $g^*_{O(p,r)}(\mathbf{x};\Lambda)$) which could maximize an objective criterion function O(p,r) of the precision (p) and recall (r) rates.

2.3 Optimal Classifier for Precision and Recall Optimization

Given the underlying distributions, $f(\mathbf{x}|\mathbf{C})$ and $f(\mathbf{x}|\overline{\mathbf{C}})$, of the feature vectors $\mathbf{x}$ in the compound class ($\mathbf{C}$) and non-compound class ($\overline{\mathbf{C}}$), it is possible to estimate the error probabilities associated with any decision region (or equivalently, any threshold, decision rules or statistical tests which could be used to define such a region) for a class. Therefore, it is possible to design the optimal classifier for some simple criterion functions if the feature distribution is very simple. In fact, procedures for designing optimal classifiers, such as the minimum error classifier, had been well studied in the speech, communication and pattern recognition communities [Devijver 82, Juang 92]. For example, the decision rule that minimizes the expected probability of classification error turns out to be a likelihood ratio test in the 2-class classification case [Devijver 82].

However, since WPR and FM are non-linear functions of classification errors (i.e., a non-linear function of n_{12} and n_{21}), it is hard to find a simple analytical discrimination function $g^*_{O(p,r)}(\mathbf{x};\Lambda)$ for testing whether an n-gram is a compound, such that the joint performance O(p,r) is maximum. Therefore, a two stage optimization scheme is proposed here in order to optimize a user specified criterion function of precision and recall, while retaining a small error rate. In the first stage, a minimum error classifier, $g^*_e(\mathbf{x};\Lambda)$, (which satisfies the minimum error criterion) is used as the base classifier to minimize the error rate (e) of classification. In the second stage, a learning method is applied, starting from the minimum error status, to optimize a user-specified criterion function of the recall and precision rates by adjusting the parameters of the classifier according to mis-classified instances.

Figure 1 shows the block diagram for training such a classifier. In the training flow, the n-grams in the training text corpus are extracted and manually inspected; those real compounds within the text corpus are used to construct a compound dictionary. The feature vectors associated with the n-grams are divided into the compound and non-compound classes according to the compound dictionary. The parameters for the compound class (Λ_c) and non-compound class ($\Lambda_{\overline{c}}$) are estimated from the distributions of the two classes. The training n-grams are then classified by the minimum error classifier. The result is compared with the compound dictionary afterward. Those misclassified n-grams are then used to adjust the parameters iteratively so that the criterion function is maximized.

The first optimization stage serves to determine the appropriate thresholds (or, more precisely, the decision boundaries) in the feature space so that as little misclassification is attained as possible. In this way, the precision and recall are expected to be improved indirectly. The second stage, on the other hand, adjusts the parameters of the classifier to achieve a local optimum of the joint precision-recall performance, starting from the minimum error status, instead of optimizing the precision and recall from arbitrary decision boundary. In other words, we are not trying to find some simple analytical discrimination function which are capable of identifying the optimal decision boundaries for precision-recall optimization. Instead, we first

establish reasonably optimized decision boundaries by using the simple discrimination function for the minimum error classifier, and then modify the decision boundaries by changing the parameters of the distribution functions of the minimum error classifier to maximize the joint precision-recall performance.

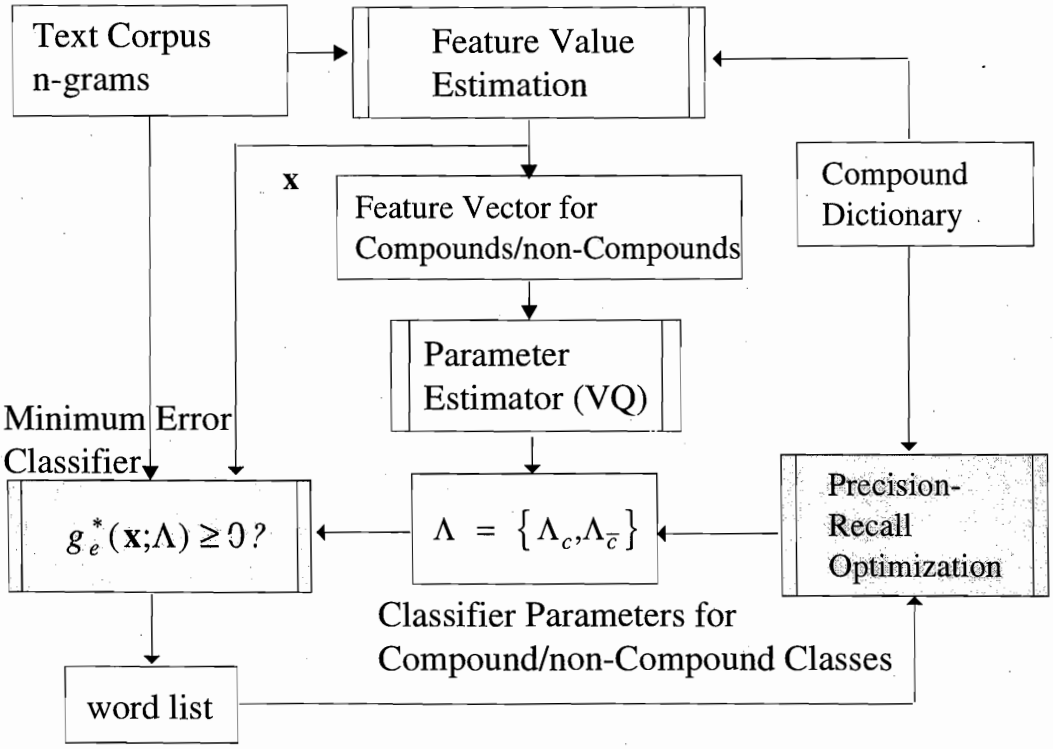


Figure 1 Supervised Training of Classifier Parameters for Precision-Recall Maximization.

The minimum error classifier is adopted at the first optimization stage since reducing classification error, in general, will improve precision and recall. In addition, it is relatively easy to implement a minimum error classifier [Devijver 82, Juang 92], and it is believed that a better local optimum could be found near the minimum error status. To see the relationship between the error rate and the precision/recall rates, first note that $p = \left(1 + n_{21}/n_{11}\right)^{-1}$ and $r = \left(1 + n_{12}/n_{11}\right)^{-1}$. The precision and recall can thus be improved by reducing n_{21} and n_{12}, respectively. Since, the error rate is propotional to $n_{12} + n_{21}$, the minimum error rate (i.e., minimum $n_{12} + n_{21}$) status is a good initial point for further optimizing the precision, recall or WPR performance. As far as F-measure is concerned, it is easy to prove that maximizing the F-measure is equivalent to minimizing $\left(n_{12} + n_{21}\right)/n_{11}$ ([Chang 97b].) Therefore, it is also appropriate for using minimum error as the criterion of the first optimization stage. In fact, if we plot the WPR (or FM) graph as a function of n_{21} and n_{12}, moving toward minimum error tends to have higher WPR (or FM) in general.

There are several issues related to the design of the minimum error classifier. As mentioned previously, it is desirable to use features that encode syntactic information, such as parts of speech, in the feature set to reject highly associated non-compound candidates; it is also desirable to use multiple features jointly to enjoy all the information contained in the features. However, the feature correlation among the features must be carefully handled in order to model the distributions of the features properly. Redundant or

non-discriminative features should be removed when combining the features. Furthermore, the parameters must be estimated in a way as to minimize the classification errors. These issues will be addressed in the following sections. Due to the length limitation, we will focus ourselves on the designing issues of the minimum error classifier in this paper. Interested readers on the precision-recall optimization techniques at the second learning stage are referred to [Chang 97b].

3. The Minimum Error Classifier for Compound Extraction

A likelihood ratio test method, which was proved to be the most powerful test [Papoulis 90], can be used as the baseline classifier to achieve minimum classification error if the distributions of the feature vectors of the two classes are known. In fact, it implicitly implies the use of the optimal thresholds (or decision boundaries) which minimizes the misclassification costs in Bayesian decision points of view [Devijver 82] if the cost for each misclassification is unity. In other words, it minimizes the probability of errors for two classes of known distributions.

To identify whether an n-gram is a compound or a non-compound, each n-gram is associated with a feature vector $\mathbf{x}$, it is then judged to see whether it is more likely to be generated from the compound class $\mathbf{C}$ or the non-compound class $\overline{\mathbf{C}}$ based on the following (log-)likelihood ratio:

$$\lambda = \frac{f(\mathbf{x}|\mathbf{C})P(\mathbf{C})}{f(\mathbf{x}|\overline{\mathbf{C}})P(\overline{\mathbf{C}})}$$

$$g(\mathbf{x}) \triangleq \log \lambda = \log f(\mathbf{x}|\mathbf{C}) - \log f(\mathbf{x}|\overline{\mathbf{C}}) + \log P(\mathbf{C}) - \log P(\overline{\mathbf{C}}),$$

where $P(\mathbf{C})$ and $P(\overline{\mathbf{C}})$ are the prior probabilities of the two classes and $f(\mathbf{x}|\mathbf{C})$ (resp. $f(\mathbf{x}|\overline{\mathbf{C}})$) is the probability density function of the feature vector $\mathbf{x}$ in the compound (resp. non-compound) class. If the likelihood ratio $\lambda \geq 1$ (i.e., the discrimination function $g(\mathbf{x})$, or the log-likelihood ratio, $\log \lambda \geq 0$) for an n-gram, then it is classified as a compound; otherwise, it is classified as a non-compound. The model parameters for the two classes are referred to as Λ_c and $\Lambda_{\overline{c}}$. They correspond to the means, variances, prior probabilities, etc. (depending how the density functions are formulated), in the above formula.

4. Features for Compound Extraction

The performance upper bound of the classifier depends on the distribution of the input feature vector $\mathbf{x}$. Many statistical features are used in various applications. In particular, the *normalized frequency* (NF) [Wu 93, Su 94b] of an n-gram, the *mutual information* (MI) [Church 90, Su 91, Chang 95] among the words within an n-gram, the *dice* metric (D) [Smadja 96, Chang 97b] among the words of the n-gram, the *contextual entropy* (H) [Tung 94, Chang 95, Chang 97a] of the neighboring words of the n-grams, are used in the classification task. (The definitions of such association features and their extension are given in the Appendix.) In addition, we will introduce a *part of speech discrimination* metric (Dpos) in this paper; it is proposed in this paper to encode syntactic information so that syntactic information could be integrated with other simple word association metrics in a simple and effective way, without resorting to complicated syntactic processing. When all such features are used, we will have the following discrimination function:

$$g(\mathbf{x}) = \log \frac{f(NF,MI,D,H,D_{pos}|\mathbf{C})P(\mathbf{C})}{f(NF,MI,D,H,D_{pos},|\overline{\mathbf{C}})P(\overline{\mathbf{C}})}$$

Since such features might contain redundant information, only a subset of the features will be automatically

selected with a feature selection mechanism for classification.

4.1 POS Discrimination (Dpos)

Part of speech (POS) is an important syntactic feature for extracting compounds. For instance, many compound words are associated with the part of speech patterns: {noun, noun}, {adjective, noun} (for bigrams) or {noun, noun, noun}, {adjective, noun, noun} (for trigrams). Previous frameworks [Wu 93, Su 94b] show that simple word association metrics are useful for extracting highly associated compound words. However, many non-compound n-grams, like 'is a', which have high association and high frequency of occurrence are also recognized as compounds. Such n-grams could be rejected if syntactic information is available.

One way to use the POS information is to measure how similarly the candidate and the compound class are tagged with different POS patterns. For instance, if the compound words are tagged as {noun, noun} in 80% of the cases and as {adjective, noun} in 20% of cases, then a candidate which was tagged as {noun, noun} and {adjective, noun} in most of the cases is very likely to be a compound. In this paper, we thus suggest the following *POS Discrimination* metric for measuring the similarity or distance between a compound word candidate x_i and the compound word class, in terms of their tagged POS patterns. The *discrimination* metric [Blahut 87] is defined as follows in terms of the distribution P_{ij} of the POS patterns of the candidate and the distribution P_j of the POS patterns of the compound class:

$$D_{pos}(x_i;\{P_{ij}\},\{P_j\}) = \sum_j P_{ij} log \frac{P_{ij}}{P_j}$$

$$P_{ij} \equiv P(j|w_i), \quad P_j \equiv P(j)$$

where P_{ij} is the probability for the *i*th compound word candidate (or n-gram) to be tagged with the part of speech pattern j (such as a {noun, noun} tag pair) and P_j is the probability for any compound word to be tagged as j.

Intuitively, the log-likelihood ratio of P_{ij} over P_j indicates how close or similar (in terms of probability of occurrence) the particular POS pattern j is, in comparison with the probability for the whole class. If the two probabilities are nearly identical, that is, $P_{ij} \approx P_j$, the log-likelihood ratio will be close to zero. Otherwise, the 'distance' will be large. The probability P_{ij} preceding the log-likelihood ratio is a weighting factor indicating how often such a 'distance' is observed; the discrimination metric is thus the expected distance between the two probability distributions of POS tagging patterns. When a compound word candidate has exactly the same distribution as the distribution for the compound class ($P_{ij} = P_j$ for all j), the 'distance' will be exactly zero. Therefore, we can gather the POS distributions of the n-grams, and use the distributions of such a distance measure in the two classes to see whether the candidate comes from the compound class.

Since this metric assumes continuous values, the distribution of this metric can be expressed in a parametric form and the parameters of the probability density functions can be estimated from a training corpus. We can thus easily incorporate such POS information for identifying compound terminologies with

a few such parameters in a very simple and effective way.

5. Experiment Environments

To investigate the various models, a corpus of 23,016 sentences (188,267 words) is prepared. The corpus is collected from a technical manual for cars. It is first processed by a morphological analyzer to normalize every word into its stem form, instead of its surface form, to reduce the number of possible variants. Since parts of speech are used as a compound extraction feature, the text is tagged by a discrimination oriented probabilistic lexical tagger [Lin 92, Lin 95] in advance. The corpus is then divided into two parts; 90% of the sentences (i.e., 20,715 sentences, 169,237 words) are used as the training corpus, and the remaining 10% (2,301 sentences, 19,030 words) are used as the testing set.

According to our experience in machine translation, most interested compounds are of length 2 or 3. Longer compounds only constitute a small fraction of interested compounds; and such long compounds can be extended by slightly modifying the definition for some association metrics. Hence, only bigrams and trigrams compounds are investigated in the current work. The corpus is therefore scanned from left to right with the window sizes 2 and 3. The lists of bigrams and trigrams thus acquired then form the lists of compound candidates of interest.

All bigrams and trigrams are submitted to three independent lexicographers of a local MT-based service translation center. The lexicographers inspect all n-grams and decide which n-grams should be considered as compounds and entered into the compound dictionary for the MT system. When there is inconsistency among their choices, the lexicographers will negotiate for a compromise. The final candidates are then used as the standard for evaluating the performance of the proposed compound extraction method. Since all the bigrams and trigrams are scanned for qualification before any experiment is conducted, the performance will reasonably reflect the performance against human judgement, the criterion for including an n-gram or not will thus not be biased by the algorithm designer's intention to have high performance.

The parameters for the compound model Λ_c and non-compound model $\Lambda_{\overline{c}}$ are evaluated from the above-mentioned training corpus, which is tagged with parts of speech and normalized into stem forms. The n-grams in the training corpus are further divided into two classes. The compound class comprises the n-grams in the compound dictionary, which was constructed by the lexicographers as described above; and the non-compound class consists of the remaining n-grams which are not in the compound dictionary. However, n-grams that occur only once or twice are excluded from consideration because such n-grams rarely introduce inconsistency and the estimated feature values are highly unreliable.

For each class, the means and standard deviations of the mutual information, normalized frequency, dice metric, contextual entropy and POS discrimination are estimated. The outlier entries (outside the range of 3 standard deviations from the mean) are discarded before estimating the model parameters so that the estimated parameters are more robust.

6. Baseline Models

To achieve minimum error classification, several factors must be carefully considered, including the features to be used, the model for formulating the underlying probability density functions of the two-classes, and the estimation to the parameters of the density functions. In the simplest form, only one feature is used for classification, and the probability density function is assumed to be a normal distribution. We then have the following baseline models:

$$\lambda = \frac{f(X_i|\mathbf{C})P(\mathbf{C})}{f(X_i|\overline{\mathbf{C}})P(\overline{\mathbf{C}})}$$

where X_i refers to any of the features among normalized frequency (NF), mutual information (MI), dice metric (D), contextual entropy (H) and POS discrimination (Dpos). Such baseline models are used to evaluate the performance for the individual feature; they will also be compared with other more complicated models to justify our proposals.

The following table gives the performance using only one feature. The shaded areas highlight the error rate performance, which is the optimization criterion at the current stage. The features are arranged in increasing order of error rates for bigrams.

		Training Set					Testing Set				
Feature		Dpos	MI	H	NF	D	Dpos	MI	H	NF	D
2-gram Baseline	Recall	11.09	0.0	4.87	6.01	12.33	8.07	0.0	1.35	2.69	36.77
	Precision	100.0	*	30.92	30.69	37.07	100.0	*	23.08	33.33	57.75
	Error Rate	11.03	12.41	13.15	13.34	13.47	21.20	23.06	23.78	23.68	20.79
	WPR(1:1)	55.54	*	17.90	18.35	24.70	54.03	*	12.22	18.01	47.26
	F-measure	19.97	*	8.41	10.05	18.50	14.93	*	2.55	4.98	44.93
Feature		Dpos	MI	H	NF	D	Dpos	MI	H	NF	D
3-gram Baseline	Recall	0.0	0.0	13.99	10.20	7.58	0.0	0.0	12.07	3.45	39.66
	Precision	*	*	42.11	22.58	25.49	*	*	58.33	66.67	41.07
	Error Rate	4.95	4.95	5.21	6.18	5.67	11.51	11.51	11.11	11.31	13.49
	WPR(1:1)	*	*	28.05	16.39	16.54	*	*	35.20	35.06	40.37
	F-measure	*	*	21.00	14.05	11.69	*	*	20.00	6.56	40.35

Table 1 Error Rate Performance Using only One Feature
(*: undefined, i.e., all candidates are classified as non-compound.).

The error rates are in the ranges of 11.03%-13.47% and 20.79%-23.78% for bigrams in the training set and the testing set respectively; for 3-grams the error rates are in the ranges of 4.95%-6.18% (training set) and 11.11%-13.49% (testing set); such performance corresponds to accuracy rates of 87-89% (76-79%) and 94-95% (87-89%) in classifying the bigram and trigram training (testing) set. Using the minimum error classifier thus achieves moderately low error rates both for the training set and testing set, without resorting to arbitrary thresholding.

Initially, however, the precision and recall are not sufficiently high except for the bigram POS discrimination case since the classifier tends to recognize most n-grams (or even all n-grams) as non-compounds. The 0% recalls and undefined precisions (designated as '*') in the table are the results of classifying all entries as non-compound as suggested by the assumed normal distributions. Such initial precisions and recalls are not a critical problem at the current stage where minimization of error counts is the major goal. It will be shown in later sections that, by incorporating more features, the error rates will be further reduced and the precision and recall will be indirectly improved toward high precision and moderate recall.

There are several problems to achieve the minimum error criterion by using the above baseline models.

First of all, various features are not used jointly to supplement each other so as to reduce the error rate. Second, the distributions are not necessarily normal for some features. (For instance, the normalized frequency is more likely to have an exponential distribution. Fortunately, the comparison between the baseline models and other more complicated models in the current work will not be affected significantly, since we actually get almost the same error rates for such features by using the exponential distribution assumption.) To resolve such problems, we will propose some methods in the following sections to improve the error rate performance further so that the first optimization stage is better conducted.

7. Feature Integration and Optimal Feature Selection

7.1 Integration of the Features

While each of the above features provides moderately good initial error rate performance in the above baseline models, it is known that jointly considering all the features would, in general, achieve better performance. It is also known that step-by-step filtering approaches, which were commonly used in traditional extraction tasks, tend to raise the precision rate at the cost of lowing the recall, since a filtering module may filter out potential candidates without using all available information; it is then not likely to acquire the global optimal precision and recall achievable by using such features. Using all features jointly in one step for optimizing the extraction task is thus emphasized here, instead of using the multiple features step-by-step in multiple filtering modules.

However, increasing the number of features may increase the modeling complexity of the classifier [Devijver 82] without increasing much performance, since some of the features might be highly correlated, and thus much redundant information will be contained in the whole set of features. Therefore, an automatic mechanism for choosing the right features is proposed here, so that only a subset of the most discriminative features are used for efficient computation without losing discrimination power.

Since our goal is to minimize the error rate performance, our strategy for finding the best feature set is to combine the current feature set (which is initially empty) with each feature not in the current feature set for conducting the likelihood ratio test. The feature which enable the classifier to minimize the error rate performance, when jointly considered with the current set of optimal features, is then added to the optimal set of features. This process starts from the baseline models and stopped when the inclusion of new feature do not improve the training set performance further. This strategy can be characterized as a kind of sequential forward selection (SFS) in the literature [Devijver 82].

7.2 Optimization Using Independent Normal Model

The performance of the classifier will also depend on how good the density function of the features fits the real training data, in addition to the feature set being used. In the simplest model, the joint probability of the features is approximated as the product of the probabilities for the individual features (by assuming that they are mutually independent), and each feature is assumed to be normally distributed. The corresponding log-likelihood ratio then becomes:

$$\log\lambda = \sum_{i=1}^{D}\left[\log f\left(x_i|\mathbf{C}\right) - \log f\left(x_i|\overline{\mathbf{C}}\right)\right] + \left[\log P(\mathbf{C}) - \log P\left(\overline{\mathbf{C}}\right)\right]$$

where the summation is taken over all features being used, and D is the dimension of the feature vector. In other words, all features are assumed to be independent in such a simplified model. With such assumptions, uncorrelated (complementary) features are likely to be included earlier than highly correlated features since

features with smaller correlation coefficients tend to be closer to the independent assumption and are likely to have better performance. The mechanism can thus select the most useful and complementary features automatically and leave redundant features unused. Table 2 shows the performances for using different numbers of features, which are selected, in sequence, by the automatic feature selection method described in the previous section, using independent normal assumption.

By applying the feature selection mechanism over all the features, the Dpos (discrimination), H (entropy), MI (mutual information), NF (normalized frequency) and D (dice) features are selected in sequence for bigrams; on the other hand, the best feature sequence for trigrams, under the current model, is Dpos, MI, H, D, NF. The SFS strategy results in the following error rate performance, where the features are arranged in the same order as the sequence in the feature selection process. For instance, the second column of the bigram performance table shows that the error rate is 8.07% when the entropy feature, H, is added to the feature set with other preceding features (in this case, the discrimination feature, Dpos).

		Training Set					Testing Set				
Feature Sequence		Dpos	H	MI	NF	D	Dpos	H	MI	NF	D
2-gram	Recall	11.09	40.41	54.61	35.34	31.30	8.07	35.43	60.54	33.63	50.67
	Precision	100.0	88.04	77.39	71.04	49.67	100.0	89.77	92.47	82.42	66.47
	Error Rate	11.03	8.07	7.61	9.81	12.46	21.20	15.82	10.24	16.96	17.27
	WPR(1:1)	55.54	64.23	66.00	53.19	40.49	54.04	62.60	76.51	58.03	58.57
	F-measure	19.97	55.39	64.03	47.20	38.40	14.93	50.81	73.17	47.77	57.50
Feature Sequence		Dpos	MI	H	D	NF	Dpos	MI	H	D	NF
3-gram	Recall	0.0	14.29	33.53	29.45	26.24	0.0	17.24	44.83	56.90	48.28
	Precision	*	100.0	70.99	46.98	33.83	*	100.0	86.67	49.25	47.46
	Error Rate	4.95	4.24	3.97	5.14	6.19	11.51	9.52	7.14	11.71	12.10
	WPR(1:1)	*	57.15	52.26	38.22	30.04	*	58.62	65.75	53.08	47.87
	F-measure	*	25.01	45.55	36.20	29.56	*	29.41	59.09	52.80	47.86

Table 2 Error rate performances of the independent normal model.

The shaded areas highlight the error rate performance, which is the optimization criterion at the current stage. The parts of speech discrimination is selected first in the two feature sequences, since the parts of speech information provide the best error rate performance among all using the normal assumption. For the bigram case, the error rate is reduced by 26.8% (from 11.03% to 8.07) when the contextual entropy information, H, is included. The inclusion of the the mutual information further reduces the error rate performance to 7.61%, corresponding to a reduction of 5.7% of the remaining errors. For trigrams, the error rates are improved slightly from 4.95% to 4.24% to 3.97 when the second and the third features (i.e., MI and H) are included, corresponding to the error reduction rates of 14% and 6%, respectively.

In addition to the improvement in error rate performance, the extra features do improve the precision and recall performance (WPR or FM, or both) as well. Although the error rate is only slightly improved (and the system retains essentially the same low error rates), the precision and recall performance is shifted away from the initial low precesion and recall status significantly. Such observations partially justify our two-stage arguments to optimize the precision and recall performance starting from a minimum error status.

However, it fails to further improve the error rate performance as the feature dimension increases

further, since the mutually independent assumption for the joint density function becomes harder and harder to be true as the feature dimension increases. For instance, the dice metric (D) and mutual information (MI) has a high correlation coefficient of about 0.6 in bigrams and 0.4 in trigrams. Another example would be the NF (normalized frequency) and H (contextual entropy), which have correlation coefficients of about 0.4-0.5 in the bigram and trigram data. The problem is resolved by considering the feature correlation and using better density functions to approximate the joint distribution as follows.

7.3 Model Refinement with Mixture of Gaussian Density Function

There are two sources of errors for including new features in the previous model, which assumes that the features are *mutually independent* and *normally distributed.* First, the independent assumption might not be true for some feature pairs. In fact, the correlation matrices for the features indicate that some of the features are highly correlated. Therefore, it is desirable to use a multivariate normal (i.e., Gaussian) distribution [Roussas 73, Rabiner 93], which encode feature correlations with a covariance matrix, to consider the effects of the correlations among the features. Second, the distributions of some features are not similar to a normal distribution. Therefore, using a mixture of the multivariate normal distribution would be a better way to fit the density functions. By increasing the number of mixtures, it is possible, in theory, to fit the shapes of the real distributions better, and thus have better estimation on the likelihoods of the joint feature vectors.

7.3.1 Using Multivariate Gaussian with Fixed Number of Mixtures

To fit the training data into a mixture of multivariate Gaussian distribution, we must estimate the means and co-variances of each mixture or cluster. The clusters are acquired using a standard vector quantization (VQ) technique [Duda 73]. For a K-mixture distribution, the feature vectors are clustered into K clusters; the mean vectors, covariance matrices and prior probabilities of the clusters are then estimated from the clustering results.

Since the number of mixtures for the underlying distributions of the joint features of various dimensions are not known, we fixed the number of mixtures (K) throughout the whole feature selection process to find the best performance. The cases for fixing K=1, 2, 3 are tried in order to find the best number of mixtures to use. The best results for 2-grams and 3-grams are given in the Tables 3-4. The comparison between the independent normal model and the K-mixture multivariate normal model (using fixed K throughout the feature selection process) is summarized in Table 5.

		Training Set					Testing Set				
Feature Sequence		Dpos	H	MI	NF	D	Dpos	H	MI	NF	D
2-gram	Recall	69.84	71.50	71.61	50.67	51.71	69.06	71.30	69.96	67.26	47.09
	Precision	100.0	97.87	88.93	62.93	45.53	100.0	95.78	93.41	80.65	52.24
	Error Rate	3.74	3.73	4.63	9.82	13.67	7.14	7.34	8.07	11.27	22.13
	WPR(1:1)	84.92	84.69	80.27	56.80	48.62	84.53	83.54	81.68	73.95	49.66
	F-measure	82.24	82.63	79.34	56.14	48.42	81.70	81.75	80.00	73.34	49.53

Table 3 The Best Bigram Performance of the Minimum Error Rate Classifier Using a 2-Mixture Multivariate Normal Density Function (K=2).

Feature Sequence		Dpos	H	MI	D	NF	Dpos	H	MI	D	NF
3-gram	Recall	63.27	68.22	67.06	51.90	54.23	75.86	74.14	74.14	36.21	37.93
	Precision	100.0	95.12	90.91	80.91	39.08	100.0	97.73	95.56	95.45	41.51
	Error Rate	1.82	1.75	1.96	2.99	6.45	2.78	3.17	3.37	7.54	13.29
	WPR(1:1)	81.63	81.67	78.98	66.40	46.65	87.93	85.93	84.85	65.83	39.72
	F-measure	77.50	79.45	77.18	63.24	45.43	86.27	84.32	83.50	52.50	39.64

Table 4 The Best Trigram Performance of the Minimum Error Rate Classifier Using a 3-Mixture Multivariate Normal Density Function (K=3).

N		Training Set					Testing Set				
	Model && Features	P	R	E	WPR	FM	P	R	E	WPR	FM
2	IN: Dpos+H	88.04	40.41	8.07	64.23	55.39	89.77	35.43	15.82	62.60	50.81
	IN: Dpos+H+MI	77.39	54.61	7.61	66.00	64.03	92.47	60.54	10.24	76.51	73.17
	Mx: Dpos+H (K=2)	97.87	71.50	3.73	84.69	82.63	95.78	71.30	7.34	83.54	81.75
3	IN: Dpos+MI	100.0	14.29	4.24	57.15	25.01	100.0	17.24	9.52	58.62	29.41
	IN: Dpos+MI+H	70.99	33.53	3.97	52.26	45.55	86.67	44.83	7.14	65.75	59.09
	Mx: Dpos+H (K=3)	95.12	68.22	1.75	81.67	79.45	97.73	74.14	3.17	85.93	84.32

Table 5 Comparison between Independent Normal (IN) Model and K-mixture Multivariate Normal (Mx) Model. (2: 2-gram, 3: 3-gram, P: Precision, R: Recall, E: Error Rate, WPR: Weighted Precision/Recall with equal weights, FM: F-measure.)

For bigram compound word detection, the best (training set) error rate performance is found in Table 3 when Dpos (parts of speech discrimination) and H (contextual entropy) are used jointly using a 2-mixture multivariate (bivariate) normal density function. The best feature sequence is identical to the normal independent model. In this case, the error rate, 3.73%, is only about 49% of the best normal independent model (using Dpos, H and MI), whose error rate is 7.61%. The WPR is also significantly improved from 66.00 to 84.69, and the FM from 64.03 to 82.63. The precision and recall for this case are 97.87% and 71.50%, respectively.

Trigram compound detection also acquires the best results by using Dpos and H, but with a 3-mixture multivariate normal density function (Table 4). The error rate is 1.75% in this case, which is only 44% of its counterpart using the independent normal model, i.e., 3.97% (using Dpos, MI and I). The results demonstrates that using a mixture of multivariate normal density function to include the correlation and fit the density function of the training data does reduce the error rate and improve the precision, recall, WPR and FM significantly.

Again, the WPR and FM are, in general, improved when the error rate is reduced. However, the tables indicate that the error rates do not decrease monotonically as the number of features are increased for a given K; the error rate decrease only for the first two or three features in the feature sequence. Besides, the error rates do not decrease monotonically either when the number of mixtures increased when comparing the performance for a specific number of features. There are several possibilities which make the fitting of the training data to a K-mixture D-variate density function imperfect in the above process; the performance thus is not monotonically increased with K or D [Chang 97b].

In particular, the number of mixtures for the underlying density function of the joint features may not

be characterized by a small K when the number of features increases to some extent. In fact, it is known in statistical pattern recognition community [Devijver 82] that when the number of features increases, the best number of mixtures for modeling the joint distribution of the features, in general, will increase quickly. For instance, two features, each having two normal mixtures, when considered jointly, may have as many as four mixtures if they are independently distributed. The number of mixtures tends to grow exponentially with the number of features in the worst cases. As a result, the real K may far exceed our searching range (K=1-3) when new features are included.

7.3.2 Improvement by Searching for the Best Number of Mixtures

The above identified problems in using a fixed number of mixtures *throughout* the whole feature selection process indicates several ways to improve the error rate performance. The simplest way would be to set an upper bound, Kmax, and tries all K <= Kmax *during* the feature selection process for each feature dimension. We thus tries several Kmax and find the best K (K*) for such searching ranges. The following table shows the results when Kmax=3.

The numbers in the parentheses indicate the best number of mixtures (K*) used. For instance, the Dpos(.)-H(2) feature sequence means that a local optimal is found when Dpos and H are jointly considered using 2-mixtures.

		Training Set					Testing Set				
Feature Sequence		Dpos(2)	H(2)	MI(3)	NF(3)	D(1)	Dpos	H	MI	NF	D
2-gram	Recall	69.84	71.50	72.12	67.05	32.12	69.06	71.30	70.40	65.92	44.39
	Precision	100.0	97.87	90.74	83.70	56.78	100.0	95.78	94.01	93.63	68.28
	Error Rate	3.74	3.73	4.37	5.71	11.45	7.14	7.34	7.86	8.89	17.58
	WPR(1:1)	84.92	84.69	81.43	75.37	44.45	84.53	83.54	82.21	79.77	56.34
	F-measure	82.24	82.63	80.37	74.46	41.03	81.70	81.75	80.51	77.37	53.80
Feature Sequence		Dpos(3)	H(3)	MI(3)	D(3)	NF(1)	Dpos	H	MI	D	NF
3-gram	Recall	63.27	68.22	67.06	51.90	24.49	75.86	74.14	74.14	36.21	44.83
	Precision	100.0	95.12	90.91	80.91	33.60	100.0	97.73	95.56	95.45	48.15
	Error Rate	1.82	1.75	1.96	2.99	6.13	2.78	3.17	3.37	7.54	11.90
	WPR(1:1)	81.63	81.67	78.98	66.40	29.04	87.93	85.93	84.85	65.83	46.49
	F-measure	77.51	79.45	77.19	63.24	28.34	86.27	84.32	83.50	52.50	46.43

Table 6 The Performance of the Minimum Error Rate Classifier
Using Multivariate Normal Density Function up to 3 Mixtures (Kmax=3).

Table 6 demonstrates that, by searching for the best K in [1, Kmax] for each feature dimension, the error rate performance is always better than (or identical to) its counterpart in Tables 3-4 of the same number of features. This justify our arguments that K must be searched for a local optimum instead of using a fixed number of mixtures all the time.

Table 6, however, still do not show monotonic decreasing of the error rates when the number of features are increased. In fact, the error rates no more decrease after the third feature is included, just like Tables 3-4. The problem is that Kmax=3 is still too small to search for a better performance even with only 3 features. In fact, we could further enlarge the searching range Kmax, and it is demonstrated in [Chang 97b] that the training set error rates for any given number of features do decrease monotonically as the searching range [1,

Kmax] is increased. We can thus expected that the error rate will decrease monotonically as the number of features are increased if we allow a much larger searching ranges. In the current task, however, it is observed that the best number of features even for 3 features are more than ten (i.e., K*>10). This would requires a very lengthy time to converge. Furthermore, the features at the tail of the feature are highly correlated with features at the front of the list, which means that they may provide little additional information once those features selected earlier are used for classification. Improving the estimation of the density functions for including such features thus would not likely to produce significant improvement. Therefore, compromise must be taken between modeling complexity and computation costs.

With Kmax=3 and two features (in which the training set error rates are minimal), we actually have testing set WPR performance of 84% (bigram) and 86% (trigram); the F-measures are about 82% and 84% for bigram and trigram, respectively.

Given the above error rate performance, it is still possible to further improve the error rate performance and thus indirectly improve the precision and recall rate performance. However, such approaches do not guarantee to get the best joint precision-recall performance, since the minimum error rate criterion, eventually, is not equivalent to maximum precision-recall. Therefore, optimizing the precision and recall performances by adjusting the parameters of the classifiers afterward is desirable. Such optimization issues and the resultant improvement, however, is beyond the scope of the current paper. Interested readers are referred to [Chang 97b].

8. Concluding Remarks

Most simple mechanisms for terminology extraction rely on trial-and-error to setup empirical thresholds for each available features, and use such features to filter out inappropriate candidates step-by-step using one feature per step. Such simple filtering-and-thresholding approaches cannot automatically optimize a user specified criterion function of precision and recall. To resolve such optimization problems, a two-stage optimization scheme is proposed. In the first stage, the system tries to reach minimum classification error to optimize the precision and recall performance indirectly, by using a two-class classifier with a likelihood test method. In the second stage, an adaptive learning method is then applied to directly optimize a criterion function of precision and recall; such a criterion function can be pre-specified by a lexicographer based on the preference over the precision and recall performance. Optimization through error rate minimization in the first stage, in particular, is addressed in detail in this paper.

The method proposed in this paper integrates mutual information, normalized frequency, dice, contextual entropy and part of speech information as the features for discriminating compounds and non-compounds. The POS discrimination metric, in particular, is proposed in the current work for encoding the syntactic constrains over possible compound candidate. Syntactic constraints can thus be easily integrated quantitatively for jointly optimizing the system performance with other word association metrics.

To reach minimum error rate in the first optimization stage, all association features are jointly considered so that all available information could be enjoyed by the system; an automatic feature selection mechanism is applied so that only the most discriminative features are used to jointly qualify compound candidates. Various models are used to fit the training data to various density functions so as to minimize the system error rate. The correlations among the features are taken into account by including the correlation matrices into the density functions, and the density functions are formulated using a mixture of multivariate

Gaussian density functions so as to well characterize the distribution of the training data.

With a training (resp. testing) corpus of 20715 (resp. 2301) sentences sampled from technical manuals about cars, the *weighted precision & recall* (WPR) using the proposed approach can achieve about 84% for bigram compounds, and 86% for trigram compounds. The F-measure performances are about 82% for bigrams and 84% for trigrams.

Appendix: Association Features for Two-Class Classification

1. Normalized Frequency (NF)

The normalized frequency for the i^{th} n-gram is defined as:

$$r_i = \frac{f_i}{\bar{f}}, \qquad \bar{f} = \frac{1}{N}\sum_{i=1}^{N} f_i$$

where f_i is the total number of occurrences of the i^{th} n-gram in the corpus, and $\bar{f}$ is the average frequency of all the entries. In other words, r_i is the normalized frequency with respect to the average frequency $\bar{f}$.

2. Mutual Information (MI)

Mutual information is a measure of word association. It is the ratio between the joint probability for a group of words to appear in the same n-gram window and the probabily for such words to occur in the same window independently. The bigram mutual information $I(x;y)$ is known as [Church 90]:

$$I(x;y) = \log_2 \frac{P(x,y)}{P(x)\times P(y)}$$

where x and y are two words in the corpus. The mutual information of a trigram is defined as [Su 91]:

$$I(x,y,z) = log\frac{P_D(x,y,z)}{P_I(x,y,z)} = log\frac{P(x,y,z)}{P_I(x,y,z)}$$
$$P_I = P(x)P(y)P(z) + P(x)P(y,z) + P(x,y)P(z)$$

where $P_D(x,y,z) \equiv P(x,y,z)$ is the joint probability for x, y, z to appear jointly as a group of words in a trigram window, and $P_I(x,y,z)$ is the probability for x, y, z to appear, independently, as a group by chance. Note that the three product terms in $P_I(x,y,z)$ correspond to three different ways in which the constituents of the trigram appear in the same trigram window by chance; $P_I(x,y,z)$ is the total probability of the various possible combinations. In general, $I(\cdot) >> 0$ implies that the words in the n-gram are strongly associated. Otherwise, their appearance as one group of words may be simply by chance.

3. Dice Metric (D)

The dice metric is commonly used in information retrieval tasks [Salton 83] for identifying closely related binary relations. The dice metric for a pair of words x, y is defined as follows [Smadja 96]

$$D_2(x,y) = \frac{P(x=1,y=1)}{\frac{1}{2}[P(x=1)+P(y=1)]},$$

where x=1 and y=1 correspond to the events that x appears in the first place and y appears in the second place of a bigram respectively. Intuitively, the dice metric is the likelihood ratio between the joint probability for

two words (or events) to occur simultaneous and the average probability for each individual word (or event to) occur in bigram pairs. Therefore, a high dice value tends to mean that x and y are highly associated.

We can also define the dice metric for triple relations following the same spirit in defining the 3-gram mutual information. However, note that in defining the bigram dice metric, the joint probability $P(x=1,y=1)$ is normalized with respect to the *average* of the marginal probabilities, $P(x=1)$ and $P(y=1)$, of the constituents instead of the *product* of the marginal probabilities (i.e., the probability of independent occurrence). Therefore, we have three different ways to normalize the joint probability with respect to the averages of the marginal constituent probabilities as follows:

$$\frac{P(x=1,\ yz=1)}{\frac{1}{2}[P(x=1)+P(yz=1)]},\ \frac{P(xy=1,\ z=1)}{\frac{1}{2}[P(xy=1)+P(z=1)]},\ or\ \frac{P(x=1,\ y=1,\ z=1)}{\frac{1}{3}[P(x=1)+P(y=1)+P(z=1)]}$$

where $P(x=1,y=1,z=1)$ is the probability that x, y and z appear simultaneously in the first, second, and third places of a trigram, $P(xy=1)$ (i.e., $P(x=1,y=1)$) is the probability that x and y appear simultaneously in the first and second places of a trigram, and $P(yz=1)$ (i.e., $P(y=1,z=1)$) stands for the probability that y and z appear in the second and third places of a trigram simultaneously. (Note that the first two normalized metrics are simply the bigram dice metrics for [x, yz] and [xy, z], respectively.) If any of the above three normalized association metrics is small, then the trigram is likely to belong to different words. Therefore, we shall use the *minimum* of the three normalized likelihood ratios to indicate the association of the trigram. The trigram dice metric is then defined as follows.

$$\begin{aligned} D_3(x,y,z) &= min\left[\frac{2P(x=1,y=1,z=1)}{P(x=1)+P(yz=1)},\frac{2P(x=1,y=1,z=1)}{P(xy=1)+P(z=1)},\frac{3P(x=1,y=1,z=1)}{P(x=1)+P(y=1)+P(z=1)}\right] \\ &\triangleq \frac{P(x=1,y=1,z=1)}{P_I'} \\ P_I' &= max\left[\frac{1}{2}[P(x=1)+P(yz=1)],\frac{1}{2}[P(xy=1)+P(z=1)],\frac{1}{3}[P(x=1)+P(y=1)+P(z=1)]\right] \end{aligned}$$

The three terms in the bracket of the *min* operator indicate three different ways in which the three words do not belong to the same lexical entry. The *min* operator means to choose the weakest evidence of association for comparison with a threshold. If the weakest evidence of association is greater than a threshold, then the trigram dice measure gives a strong indication that the three words belong to the same lexical entry. Given the above definition, only those trigrams which appear simultaneously with significantly higher probability than the maximum probability of the various other combinations of the constituents are considered compound candidates.

4. Contextual Entropy (H)

The left and right contextual entropies [Tung 94] are defined respectively as follows:

$$\begin{aligned} H_L(x) &= -\sum_{w_i} P_L(w_i;x)\log P_L(w_i;x) \\ H_R(x) &= -\sum_{w_i} P_R(x;w_i)\log P_R(x;w_i) \end{aligned}$$

where w_i is the left or right neighboring word of an n-gram x, and the probability that w_i appear as the left or right neighbor of an n-gram x is represented as $P_L(w_i;x)$ and $P_R(x;w_i)$, respectively. If the contextual

entropy is large, which means that the neighbors of x are randomly distributed, then x tends to be a lexical unit by itself; otherwise, x and w_i are likely to appear simultaneously, which implies that x is unlikely to be a lexical unit by itself. In the current work, we use the average of H_L and H_R as a single feature instead of using the two entropy metrics.

References

Appelt, Douglas E., Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson, "FASTUS: A Finite-State Processor for Information Extraction from Real-World Text," *Proc. IJCAI-93*, Chambery, France, Aug. 1993.

Blahut, Richard E., *Principles and Practice of Information Theory*, Addison-Wesley Publishing Company, MA, USA, 1987.

Bourigault, D. "Surface Grammar Analysis for the Extraction of Terminological Noun Phrases," In *Proceedings of COLING-92*, vol. 4, pp. 977--981, 14th International Conference on Computational Linguistics, Nantes, France, Aug. 23--28, 1992.

Calzolari, N. and R. Bindi, "Acquisition of Lexical Information from a Large Textual Italian Corpus," In *Proceedings of COLING-90*, vol. 3, pp. 54--59, 13th International Conference on Computational Linguistics, Helsinki, Finland, Aug. 20--25, 1990.

Chang, Jing-Shin, Yi-Chung Lin and Keh-Yih Su, "Automatic Construction of a Chinese Electronic Dictionary," *Proceedings of the Third Workshop on Very Large Corpora*, pp. 107-120, MIT, June, 1995.

Chang, Jing-Shin and Keh-Yih Su, 1997a. "An Unsupervised Iterative Method for Chinese New Lexicon Extraction", to appear in *International Journal of Computational Linguistics & Chinese Language Processing*, 1997.

Chang, Jing-Shin, 1997b. *Automatic Lexicon Acquisition and Precision-Recall Maximization for Untagged Text Corpora*, PhD dissertation, National Tsing-Hua University, Hsinchu, Taiwan, July 1997.

Chen, S.-C. and K.-Y. Su, "The Processing of English Compound and Complex Words in an English-Chinese Machine Translation System," In *Proceedings of ROCLING I*, Nantou, Taiwan, pp. 87--98, Oct. 21--23, 1988.

Chen, S.-C., J.-S. Chang, J.-N. Wang and K.-Y. Su, "ArchTran: A Corpus-Based Statis ics-Oriented English-Chinese Machine Translation System," *Proceedings of Machine Translation Su mit III*, pp. 33--40, Washington, D.C., USA, July 1--4, 1991.

Church, K.W. and P. Hanks, "Word Association Norms, Mutual Information, and Lexicography," *Computational Linguistics*, pp. 22--29, vol. 16, Mar. 1990.

Devijver, Pierre A. and Josef Kittler, 1982. *Pattern Recognition: A Statistical Approach*, Prentice-Hall Inc., N.J., USA, 1982.

Duda, Richard O. and Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, NY, USA, 1973.

Hirschman, Lynette and Marc Vilain, *Extracting Information from the MUC*, Tutorial of the ACL 95, MIT, Cambridge, MA, June 16, 1995.

Hobbs, Jerry R. "FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text," *Proc. of ROCLING IX*, pp. 199-231, Natl. Cheng-Kung Univ., Tainan, Taiwan, Aug. 1996.

Juang, B.-H. and S. Katagiri, "Discriminative Learning for Minimum Error Classification," *IEEE Trans. on Signal Processing*, vol. 40, no. 12, pp. 3043-3054, Dec. 1992.

Lin, Y.-C., T.-H. Chiang and K.-Y. Su, "Discrimination Oriented Probabilistic Tagging," In *Proceedings of ROCLING V*, Taipei, Taiwan, pp. 85--96, Sep. 18--20, 1992.

Lin, Y.-C., T.-H. Chiang and K.-Y. Su, "The effects of learning, parameter tying and model refinement for improving probabilistic tagging," *Computer Speech and Language*, vol. 9, no. 1, pp. 37-61, Academic Press. Jan. 1995.

Papoulis, A., *Probability & Statistics*, Prentice Hall, Inc., Englewood Cliffs, NJ, USA, 1990.

Rabiner, L., and B.-H. Juang. *Fundamentals of Speech Recognition.* Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1993.

Roussas, G. G., *A First Course in Mathematical Statistics*, Addison-Wesley Publishing Company, 1973.

Salton, Gerard and Michael J. McGill, *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, 1983.

Smadja, Frank, "Retrieving Collocations from Text: Xtract," Computational Linguistics, vol. 19, no. 1, pp. 143-177, 1993.

Smadja, Frank, Kathleen R. McKeown and Vasileios Hatzivassiloglou, "Translating Collocations for Bilingual Lexicons: A Statistical Approach," Computational Linguistics, vol. 22, no. 1, pp. 1-38, 1996.

Su, K.-Y., Y.-L. Hsu and C. Saillard, "Constructing a Phrase Structure Grammar by Incorporating Linguistic Knowledge and Statistical Log-Likelihood Ratio," In *Proceedings of ROCLING IV*, Kenting, Taiwan, pp. 257--275, Aug. 18--20, 1991.

Su, K.-Y.. and C.-H. Lee, 1994a, "Speech recognition using weighted HMM and subspace projection approaches," *IEEE Trans. Speech and Audio Processing*, vol. 2, no.1, pp. 69-74, Jan. 1994.

Su, K.-Y., M.-W. Wu and J.-S. Chang, 1994b. "A Corpus-based Approach to Automatic Compound Extraction", *Proceedings of ACL 94*, 32nd Annual Meeting of the ACL, pp. 242-247, New Mexico State University, 27-30 June 1994.

Tung, Cheng-Huang and Hsi-Jian Lee, "Identification of Unknown Words from a Corpus," *Computer Processing of Chinese & Oriental Languages*, Vol. 8, pp. 131-145, (*Proceedings of ICCPOL-94*, pp. 412-417, Taejon, Korea,) Dec. 1994.

Wang, Mei-Chu, Chu-Ren Huang and Keh-Jiann Chen, "The Identification and Classification of Unknown Words in Chinese: An N-Grams-Based Approach," In Ishikawa, Akira and Yoshihiko Nitta, Eds. Festschrift for Professor Akira Ikeya, pp. 113-123. Tokyo: The Logico-linguistics Society of Japan, 1995.

Wu, Ming-Wen and Keh-Yih Su, "Corpus-based Automatic Compound Extraction with Mutual Information and Relative Frequency Count", In *Proceedings of ROCLING VI*, Nantou, Taiwan, ROC Computational Linguistics Conference VI, pp. 207-216, Sep. 2-4, 1993.

Proper Name Extraction from Web Pages for Finding People in Internet

Hsin-Hsi Chen and Guo-Wei Bian

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, TAIWAN, R.O.C.

E-mail: hh_chen@csie.ntu.edu.tw; gwbian@nlg.csie.ntu.edu.tw

Abstract

This paper proposes a method to extract proper names and their associated information from Web pages for Internet/Intranet users automatically. It extracts information from World Wide Web documents, including proper nouns, E-mail addresses and home page URLs, and finds the relationship among these data. Natural language processing techniques are employed to identify and classify proper nouns, which are usually unknown words. Different kinds of clues such as spelling method, adjacency principle and HTML tags are used to relate proper nouns to their corresponding E-mail and/or URL. With the mapping schemes, the extracted information is more accurate than the results from the traditional searching engines. The results can be used as the database of the services for finding people and organizations in Internet. Such searching services are very useful for human communication and dissemination of information.

1. Introduction

With the rapid growth of Internet in recent years, World Wide Web (WWW) becomes a very large knowledge source nowadays. Much information is disseminated through the giant media. One of the problems in the large cyber space is: it is very difficult to know where an entity, which is a concrete object that can send and receive information, is. For communication purpose, we usually want to know a person's or a company's E-mail address, or his/her home page URL. Yellow pages, which are E-mail directory or URL directory in this case, can help to find the related information. Because WWW is a very large database and is created dynamically, it is hard to set up such a kind of yellow pages manually. In the other way, the current searching engines just index the contents of the Web page with their URL. Supposing a page contains many proper names, the searching engine will index all the proper nouns with this page's URL. However, only one of these proper nouns or none is the owner of the page. For the need of intelligent processing, the precision of the searching

engines is too low for such a task of finding people.

Hopefully, very large portion of WWW is composed of natural language documents that can be regarded as a text corpus. Corpus analysis techniques in natural language processing (CL, 1993) can be employed to extract knowledge from WWW. And using the semantics of the content and HTML tags, some mapping schemes are proposed to relate the knowledge with the URL of Web page.

This paper will propose a method to construct yellow pages for Internet/Intranet users automatically. It extracts information including proper nouns, E-mail addresses and home page URLs from WWW documents, and finds the relationship among these data. The problems to be tackled are shown as follows:

(1) Proper nouns, which are always unknown words, have to be identified and classified from WWW corpus. Those proper nouns that denote organizations are usually hierarchical. Such kinds of relationships must be distinguished.

(2) There may be more than one proper noun, more than one E-mail, and more than one URL in a WWW document. Thus we have to find a mapping from a set of E-mail addresses (or URLs) to a set of proper nouns.

The language models proposed in this paper are experimented on Taiwan WWW home pages. Section 2 introduces WWW documents and the semantics of the HTML annotation. The hierarchical nature and the related HTML tagging (1996) are discussed. Section 3 shows the overview of our yellow page constructor. Section 4 presents the identification algorithms for proper nouns. Here, we focus on personal names and organization names. Section 5 touches on the mapping algorithms between proper nouns and the related information. Section 6 discusses the experiments, and section 7 concludes some remarks.

2. WWW Documents

The first step in constructing yellow pages is to know where the proper nouns, E-mail addresses and URLs are in WWW documents. WWW documents are different from the traditional text corpus in that they are HTML (HyperText Markup Language) files. The tagging information provides some clues, but it also introduces some noises. How to use the information is a very important issue in applications on Internet, e.g., cross-language information retrieval (Bian and Chen, 1997). In plain text, each sentence always has a

sentence terminator such as full stop, question mark and exclamation mark. These symbols split each document into several processing units. In HTML files, these punctuation marks do not always appear. Quasi-sentences are defined according to some of HTML tags shown below:

- Title (TITLE)
- Headings (H1, H2, …, H6)
- Address (ADDRESS)
- Unordered Lists (UL, LI)
- Ordered Lists (OL, LI)
- Definition Lists (DL, DT, DD)
- Tables (TABLE, TD, TH, TR)

Besides, some punctuation symbols like '|' and ':' have the same effects. In contrast to the above sentence delimiters, the font style elements may introduce noises. Bold (B), italic (I), superscripts (SUP), subscripts (SUB) and font (FONT) can be used to emphasize some points in texts. However, these elements produce many unknown words because a word is split into several parts by HTML tags. Thus these tags should be treated as meta-information and hidden from processing.

The links denoted by anchors (A) in the WWW documents are one of the possible sources of proper nouns and the related information. Some WWW documents shown in Appendix A demonstrate their typical features. The first example is the home page of National Taiwan University (NTU, http://www.ntu.edu.tw/). The entity that we are interested in is NTU ('國立台灣大學'), which is an organization name. Underline shows a link to other home pages. The second example follows from NTU Link. The interesting entities are Offices of Academic Affairs ('教務處'), of Student Affairs ('學務處'), of Business Affairs ('總務處'); University Library ('圖書館'); Computer and Information Network Center ('計算機及資訊網路中心'); Population Studies Center ('人口研究中心'). Those units that do not have any links are not considered. For example, the home pages for Accounting Office ('會計室') and Military Instructors' Office ('軍訓室') are not constructed now, so that they are not listed in the final yellow pages. Following the link for Colleges, Schools, Departments, Graduate Institutes and Affiliated Organizations, we can retrieve more information. All these units form a hierarchical structure. A link in the HTML file may be represented as follows:

 text

When "text" is a proper noun, its home page URL may be described by "argument". Consider an example in the NTU Link home page. The link of Office of the Dean of Academic Affairs ('教務處') is shown below:

教務處
/ Office of the Dean of Academic Affairs

If the proper noun and its URL are put into yellow pages directly, this entry may be ambiguous. This is because many universities have the similar organization. Therefore we should keep the hierarchical path of the Web pages to disambiguate the meaning of a proper noun. Further, the relative URLs need to be modified as the absolute ones for keeping the complete URL information.

Besides the link field, proper nouns may appear in other positions in a WWW document. To deal with these objects is more complex. An additional algorithm is needed to associate URLs and E-mail addresses with suitable proper nouns. Different kinds of clues such as spelling method, adjacency principle and HTML tags (e.g., title, headings, address, font style elements) are employed.

3. System Overview

We periodically collect the home pages from Internet/Intranet by a spider. The yellow page constructor first analyzes these HTML files. The basic processing units (sentences or quasi-sentences) and HTML meta-information are gathered. Because a Chinese sentence (or quasi sentence) is composed of a sequence of characters without word boundaries (Chen and Lee, 1996), a Chinese segmentation system identifies the word tokens. Then, a proper noun identification system (see Section 4) extracts personal names and organization names. During processing, the information in anchor parts is placed in the anchor set (AS). Other information, i.e., that appears in non-anchor parts, is placed in one of the content sets (CSes) for the different types of information. In current implementation, there are three content sets - say, CS_Proper-Noun, CS_E-Mail and CS_HTTP. They record proper nouns, E-mail addresses and URLs, respectively. For the anchor set, the remaining task is simple. We just relate the proper noun found in an anchor to the corresponding URL. For the content sets, a mapping algorithm (see Section 5) associates URLs and/or E-mail addresses with a suitable proper noun. Algorithm 1 shows the information extraction part of the yellow page constructor.

Algorithm 1. Information Extraction

Input: An HTML file or a plain text

Output: An anchor set (AS) and three content sets (CSs)

Method:

1. [HTML Parser]
   Identify sentence boundary and collect those HTML tags that are useful for information mapping.

2. [Chinese Segmentation System]
   For each processing unit (a sentence or a quasi-sentence), identify the word boundary.

3. For each processing unit
   3.1 [Extracting the Anchor Information]
   for each anchor (Text)
   { Identify and classify proper noun (PN) within Text.
   if PN exists, add the tuple (PN, protocol://host/path) to the Anchor Set (AS)
   }

   3.2 [Extracting the Content Information]
   3.2.1 Identify and classify proper nouns (PNs)
   if found
   { add PN to CS_Proper-Noun with the following attributes:
   position information (token_no) and associated HTML meta information (<TITLE>, <Hn>, <Address>, <Bold>, <Font> and <Italic>)
   }
   3.2.2 Extract different types of information with token_no, and add to the corresponding Content Sets (CSes).

4. End

4. Identification of Proper Nouns

Proper nouns which are not collected in lexicons are major unknown words in natural language texts. Several methods (Boguraev and Pustejovsky, 1996; Chen and Lee, 1996; Mani, *et al.*, 1993; McDonald, 1993; Paik, *et al.*, 1993) have been proposed to identify proper nouns. Of these, Chen and Lee (1996) present various strategies to identify and classify three types of proper nouns in Chinese texts, i.e., Chinese personal names, Chinese

transliterated personal names and organization names. In large-scale experiments, the average precision rate is 88.04% and the average recall rate is 92.56% for the identification of Chinese personal names. The average precision rate and the average recall rate for the identification of organization names are 61.79% and 54.50%, respectively. We follow this work on the extraction of personal names and organization names from Taiwan Web pages.

4.1 Identification of Personal Names

A Chinese personal name is composed of surname and name parts. Most Chinese surnames are single character and some rare ones are two characters. A married woman may place her husband's surname before her surname. Thus there are three possible types of surnames, i.e., single character, two characters and two surnames together. Most names are two characters and some rare ones are one character. Theoretically, every character can be considered as names rather than a fixed set. Thus the length of Chinese personal names range from 2 to 6 characters. The baseline models for the extraction are shown as follows:

Model (a) Single character

$$(1)\ \frac{\#C_1}{\&C_1}\times\frac{\#C_2}{\&C_2}\times\frac{\#C_3}{\&C_3} > Threshold1$$

$$(2)\ \frac{\#C_1}{\&C_1} > Threshold2 \text{ and } \frac{\#C_2}{\&C_2}\times\frac{\#C_3}{\&C_3} > Threshold3$$

Model (b) Two characters

$$(3)\ \frac{\#C_2}{\&C_2}\times\frac{\#C_3}{\&C_3} > Threshold4$$

Model (c) Two surnames together

$$(4)\ \frac{\#C_{11}}{\&C_{11}}\times\frac{\#C_{12}}{\&C_{12}}\times\frac{\#C_2}{\&C_2}\times\frac{\#C_3}{\&C_3} > Threshold5$$

$$(5)\ \frac{\#C_{11}}{\&C_{11}}\times\frac{\#C_{12}}{\&C_{12}} > Threshold6 \text{ and } \frac{\#C_2}{\&C_2}\times\frac{\#C_3}{\&C_3} > Threshold7$$

For different types of surnames, different models are adopted. Because the surnames with two characters are always surnames, Model (b) neglects the score of surname part. Models

(a) and (c) have two score functions. They solve the problem of very high score of surnames. The above three models can be extended to the single-character names. When a candidate cannot pass the thresholds, its last character is cut off and the remaining string is tried again. Thresholds are trained from a large-scale Chinese name corpus of 219,738 Chinese personal names. We let 99% of the training data pass the thresholds.

Text provides many useful clues from three different levels - say, character, sentence and paragraph levels. The baseline model belongs to the character level. Titles, mutual information and punctuation marks come from sentence level. When a title such as "President" appears before (after) a string, it is probably a personal name. Mutual information (Church and Hanks, 1990), which provides a measure of word association, is employed to tell the difference between a name and a content word. We check the string which can serve as a name or a content word with its surrounding words. When they have a strong relationship, it has high probability to be a content word rather than a name. The punctuation marks play an important role in identification. Personal names usually appear at the head or the tail of a sentence. The last clue is the paragraph information. A personal name may appear more than once in a paragraph. We use cache to store the identified candidates, and reset cache when next paragraph is considered.

4.2 Organization Names

The structures of organization names are more complex than those of personal names. Basically, a complete organization name can be divided into two parts, i.e., name and keyword. Many words can serve as names, but only some fixed words can be regarded as keywords. Thus keyword is an important clue to extract the organization names. However there are still several difficult problems. First, a keyword is usually a common content word. It is not easy to tell their difference. Second, a keyword may appear in an abbreviated form, or even be omitted completely. Third, some organization names are very long, so it is hard to decide the left boundary.

This paper only touches on the third problem. Keywords, which are good indicators, play the similar role of surnames. They show not only the possibility of an occurrence of an organization name, but also its right boundary. Prefix is a good marker for possible left boundary. Parts of speech such as transitive verbs, adjectives, numerals and classifiers are

also useful to determine the left boundary. The name part of an organization cannot beyond these critical parts of speech. Because a tagger is not involved before identification, the part of speech of a word is determined wholly by its lexical probability. Finally, the formulation of the name part of an organization name is considered. If the word preceding a keyword is a place name or a personal name, it forms the name part of an organization. Otherwise, we use the word association model to determine the left boundary. The postulation is: the words to compose a name part usually have strong relationship. The mutual information mentioned in the last section is also used to measure the relationship of two words.

5. A Mapping Algorithm

Algorithm 2 is a mapping algorithm that relates URLs and/or E-mail addresses to the proper nouns. A score function that considers spelling method, adjacency principle and HTML tags is used to determine the relationship among proper nouns and the related information.

Algorithm 2. Information Mapping

Input: Three Contents Sets (CSs)
A Threshold and a Window_Size of context

Output: A Mapping Set (MS)

Function: Mapping CS_E-mail (CS_HTTP) with CS_Proper-Name

Method:

1. Set MS to an empty set.
2. For each CS set (i.e., CS_E-mail and CS_HTTP)

```
{   /* the mapping between CS and CS_Proper-Noun may be Many-to-one. */
    copy CS_Proper-Noun to CD
    for each entry Info in CS
    {   PN is an entry whose offset from Info is less than Window_Size
            and Score(Info, PN) is the maximum in CD.
        if Score(Info, PN) > Threshold
        {   add (Info, PN) into MS
        }
    }
}
```

3. End

The ranking function is defined as follows:

$$Score(\text{Info, PN}) = \left(\frac{HTML\_SCORE(\text{PN})+1}{\text{abs(Info.token\_no - PN.token\_no)}} + \frac{Title(PN)}{\text{Total\_tokens - Info.token\_no+1}}\right) + Pinyin\_Similarity(\text{PN, Info})*E\text{-}mail(\text{Info})*1.2$$

$$HTML\_SCORE(\text{PN}) = Title(PN)+Heading(PN)+Address(PN)+Bold(PN)+Font(PN)+Italic(PN)$$

where Title(), Heading(), Address(), Bold(), Font(), Italic() and E-mail() are Boolean functions.

The *Score* function combines the following heuristic rules:

1. **Spelling Method.** If the extracted information (Info) is an E-mail address, the similarity between Info and the proper noun (PN) is considered. Because user-id in E-mail address is often transliterated from Chinese name, the similarity has the highest priority than the other cues. We often adopt a Pinyin system (Lu, 1995) to transliterate Chinese name. The Pinyin Similarity is defined as follows:

   $$Pinyin\_Similarity(\text{PN, E-mail}) = \frac{\#\text{ of alphabets of user - id that match to the pinyin transliteration of PN}}{\text{total \# of alphabets in the user - id of the E - mail address}}$$

   For example, the Pinyin transliteration of "邊國維" is "Bian Guo Wei".

   Pinyin_Similarity(邊國維, gwbian@nlg.csie.ntu.edu.tw) = $\frac{6}{6} = 1$

   Pinyin_S(邊國維, arthur_bian96@nlg.csie.ntu.edu.tw) = $\frac{4}{10} = 0.4$

   Pinyin_S(邊國維, arthur@nlg.csie.ntu.edu.tw) = $\frac{0}{6} = 0$

2. **Adjacency Principle.** Proper nouns and the related information are often near. The distance between Info and PN is measured in terms of the number of intervened tokens. Recall that we assign each object a unique token number. The closer pair has a larger score.

3. **HTML Tags.** The proper nouns (PNs) that appear in Title (<Title>) / Heading(<Hn>…</Hn>) / Address, or are described by the font style (Bold,

Italic and Font tag elements) are given larger weight for ranking than other normal proper nouns.

6. Experiments

In our initial experiments, total 703 home pages are collected from our campus NTU Web (http://www.ntu.edu.tw/). The collected pages are classified into an anchor set and a content (non-anchor) set. Then, the personal names and organization names are corrected by human for evaluation. The window size (Window_Size) of context is 6 and the score threshold (Threshold) is 0.2 for the mapping algorithm. Table 1 shows the results of identification in both sets and the mapping result in the content set. Appendix B and C demonstrate some extracted examples.

Anchor Set	# of items identified by program	# of items in the home pages of NTU	# of items identified correctly by program	Precision	Recall
Personal Name	228	255	189	82.89%	74.12%
Organization Name	611	746	213	34.86%	28.55%

(a) Identification of Proper Nouns in the Anchor Set

Content Set	# of items identified by program	# of items in the home pages of NTU	# of items identified correctly by program	Precision	Recall
Personal Name	3343	1732	1470	43.97%	22.14%
Organization Name	2272	3029	503	22.14%	16.61%

(b) Identification of Proper Nouns in the Content Set

Content Set Mapping	# of items extracted by program	# of items mapped correctly by program	# of items mapped incorrectly by program	Accuracy
E-mail	64	18	5	78.26%
HTTP	16	1	0	100%

(c) The Mapping Result in the Content Set

Table 1 The Results of Identification and Information Mapping.

In anchor part, there are 6204 linking items. Of these, the numbers of personal names and organization names are 255 and 746, respectively. That is, 83.87% that are irrelevant should be screened for the task of finding people. The precision and the recall are 82.89% and 74.12% for the identification of personal names. But the precision and the recall for the identification of organization names are much lower than those in our previous work. The major errors result from the conjunctions and compounds of the organization names. For these complex proper names, the correct boundaries are not determined in identification task. Some examples of errors are shown in the following.

<A href="http://www.bp.ntu.edu.tw/">台大建築與城鄉研究所</A>
Oname: 城鄉研究所
公立大學暨獨立學院 / Public University and College
Oname: 公立大學
公立大學暨獨立學院 / Public University and College
Oname: 獨立學院
長庚醫學暨工程學院 / Chang Gung College of Medicine and Technology
Oname: 工程學院
教育網路中心 / Educational Network Center
Oname: 網路中心
華梵人文科技學院 / Huafan College of Humanities and Technolgy
Oname: 科技學院

In the other way, there are 1732 proper names and 3029 organization names listed in the content part of the 703 Web pages. Only one of these proper nouns or none is the owner of one page. At least, 85.23% of these names are irrelevant. The current searching engine will index all the proper nouns with their URLs. This is the reason why the precision of the searching engines is too low for such a task of finding people.

Totally, there are 64 E-mail addresses and 16 HTTP URLs extracted in the non-anchor part. With the mapping heuristics, 18 E-mail address are assigned the correct personal names or organization names; 5 E-mail addresses are assigned incorrectly; and the others have no associated ones. The mapping algorithm achieves 78.26% accuracy to relate the information with the proper nouns. We found the spelling Pinyin similarity provides very good heuristics to relate the E-mail addresses to the proper nouns, even they are not the nearest pairs. Some experimental data and results are shown in Appendix C.

7. Concluding Remarks

This paper proposes a computer-aided information extraction method to construct yellow pages for Internet/Intranet users or to build the database of the services for finding people and organizations in Internet. The results show much interesting information can be extracted from WWW. However, the complete identification for the conjunction and compound of the organization names need further investigations in future works. Other types of information, e.g., addresses, phone numbers, and so on, will be considered in the next step. Besides, the hierarchical relationship should be tackled to set up complete yellow pages.

References

Bian, G.W. and Chen, H.H. (1997). "An MT Meta-Server for Information Retrieval on WWW", *Working Notes of the AAAI Spring Symposium on Natural Language Processing for the World Wide Web*, Palo Alto, California, USA., March, 1997, pp.10-16.

Boguraev, B. and Pustejovsky, J. (1996). *Corpus Processing for Lexical Acquisition*, MIT Press, Cambridge, MA, USA., 1996.

Chen, H.H and Lee, J.C. (1996) "Identification and Classification of Proper Nouns in Chinese Texts," *Proceedings of 15th International Conference on Computational Linguistics*, 1996, pp. 222-229.

Church, K.W. and Hanks, P. (1990) "Word Association Norms, Mutual Information, and Lexicography," *Computational Linguistics*, Vol. 16, No. 1, 1990, pp. 22-29.

CL (1993) "Special Issues on Using Large Corpora," *Computational Linguistics*, Vol. 19, Nos. 1-2, 1993.

HTML (1996) *HyperText Markup Language*, http://www.w3.org/pub/WWW/Markup.

Lu, Suping (1995) "A Study on the Chinese Romanization Standard in Libraries," *Cataloging and Classification Quarterly*, 21, 81-97.

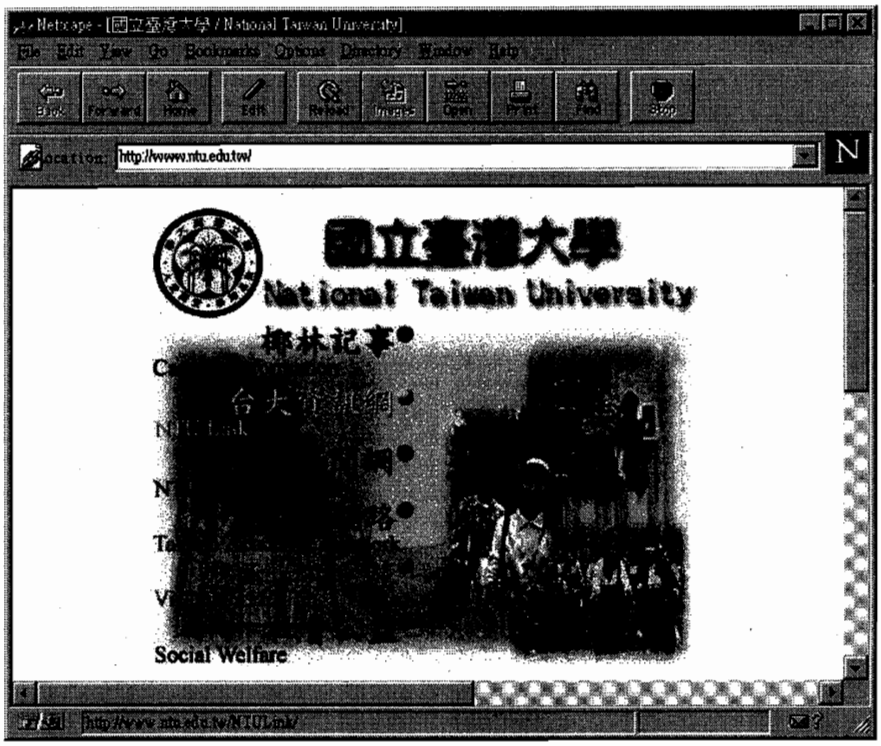
Mani, I., *et al.* (1993) "Identifying Unknown Proper Names in Newswire Text," *Proceedings of Workshop on Acquisition of Lexical Knowledge from Text*, 1993, pp. 44-54.

McDonald, D. (1993) "Internal and External Evidence in the Identification and Semantic Categorization of Proper Names," *Proceedings of Workshop on Acquisition of Lexical Knowledge from Text*, 1993, pp. 32-43.

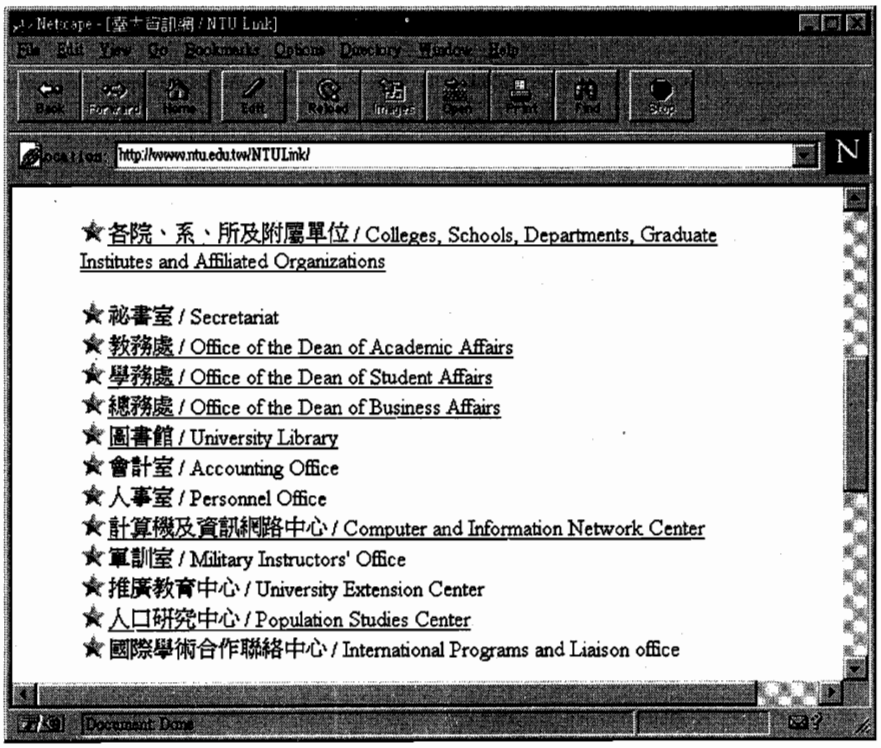
Paik, W., *et al.* (1993) "Categorization and Standardizing Proper Nouns for Efficient Information Retrieval," *Proceedings of Workshop on Acquisition of Lexical Knowledge from Text*, 1993, pp. 154-160.

Appendix A. Hierarchical Features of Home Pages

(1) Home page of National Taiwan University



(2) Home Page from NTU Link



Appendix B. Some Experimental Results in Anchor Part

In the following, Oname and Pname denote the extracted organization names and personal names respectively.

[Organization-School (Oname)]

國立臺灣大學 / National Taiwan University Oname: 國立臺灣大學
國立政治大學 / National Chengchi University Oname: 國立政治大學
國立清華大學 / National Tsing Hua University Oname: 國立清華大學
國立交通大學 / National Chiao Tung University Oname: 國立交通大學
國立臺灣師範大學 / National Taiwan Normal University Oname: 國立臺灣師範大學
國立中央大學 / National Central University Oname: 國立中央大學
國立中山大學 / National Sun Yat-sen University Oname: 國立中山大學
國立成功大學 / National Cheng Kung University Oname: 國立成功大學
國立中正大學 / National Chung Cheng University Oname: 國立中正大學
國立陽明大學 / National Yang Ming University Oname: 國立陽明大學
國立東華大學 / National Dong Hwa University Oname: 國立東華大學
國立臺灣海洋大學 / National Taiwan Ocean University Oname: 國立臺灣海洋大學
國立暨南國際大學 / National Chi-Nan University Oname: 國立暨南國際大學
中央警察大學 / Central Police University Oname: 警察大學
國立台北師範學院 / National Taipei Teachers College Oname: 國立台北師範學院
台北市立師範學院 / Taipei Municipal Teachers College Oname: 台北市立師範學院
國立藝術學院 / National Institute of the Arts Oname: 國立藝術學院
國立台北護理醫學院 / National Taipei College of Nursing Oname: 國立台北護理醫學院
國立台灣工業技術學院 / National Taiwan Institute of Technology Oname: 國立台灣工業技術學院
<A HREF="http://www.princeton.edu/index.html">普林斯頓大學</A> Oname: 普林斯頓大學
慈濟醫學院 / Tzu Chi College of Medicine Oname: 慈濟醫學院
朝陽技術學院 / Chaoyang Institute of Technology Oname: 朝陽技術學院
元智工學院 / Yuan-Ze Institute of Technology Oname: 元智工學院
高雄工學院 / Kaohsiung Polytechnic Institute Oname: 高雄工學院
中華工學院 / Chung-Hua Polytechnic Institute Oname: 中華工學院
大葉工學院 / Da-Yeh Institute of Technology Oname: 大葉工學院
國立臺北商業專科學校 / National Taipei College of Business
Oname: 國立臺北商業專科學校
國立臺中商業專科學校 / National Taichung Institute of Commerce
Oname: 國立臺中商業專科學校
國立屏東商業專科學校 / National Pingtung Institute of Commerce
Oname: 國立屏東商業專科學校
國立嘉義農業專科學校 / National Chia-Yi Institute of Agriculture
Oname: 國立嘉義農業專科學校
國立宜蘭農工專科學校 / National Ilan Institute of Agriculture and Technology
Oname: 宜蘭農工專科學校
國立高雄工商專科學校 / National Kaohsiung Institute of Technology
Oname: 國立高雄工商專科學校
國立勤益工商專科學校 / National Chinyi Institute of Technology
Oname: 國立勤益工商專科學校
國立聯合工商專科學校 / National Lien-Ho College of Technology and Commerce
Oname: 國立聯合工商專科學校
國立雲林工業專科學校 / National Yunlin Polytechnic Institute
Oname: 國立雲林工業專科學校
國立高雄餐旅管理專科學校 / National Kaohsiung Hospitality College
Oname: 國立高雄餐旅管理專科學校
國立台灣體育專科學校 / National Taiwan College of Physical Education
Oname: 國立台灣體育專科學校
臺南家政專科學校 / Tainan College of Home Economics Oname: 臺南家政專科學校
佛教慈濟護理專科學校 / Buddhist Tz'u Chi Junior College of Nursing
Oname: 慈濟護理專科學校
健行工商專校 / Chien Hsien Institute of Technology and Commerce Oname: 健行工商
萬能工商專科學校 / VanNung Institute of Technology Oname: 萬能工商專科學校
南亞工商專科學校 / Nanya Junior College Oname: 南亞工商專科學校
龍華工商專科學校 / Lunghwa Junior College of Technology and Commerce
Oname: 龍華工商專科學校

明新工商專校 / Ming Hsin Institute of Technology Oname: 明新工商
大華工商專科學校 / Ta Hua College of Technology and Commerce
Oname: 大華工商專科學校
親民工商專科學校 / Chin Min College of Technology and Commerce
Oname: 親民工商專科學校
樹德工商專科學校 / Shu Teh Junior College of Technology Oname: 樹德工商專科學校
中州工商專校 / Chung Chou Junior College of Technology and Commerce
Oname: 中州工商專校
建國工商專科學校 / Chienkuo Junior College of Technology Oname: 建國工商專科學校
吳鳳工商專科學校 / Wu-Feng Junior College of Technology and Commerce
Oname: 吳鳳工商專科學校
南台工商專科學校 / Nan Tai College of Technology and Commerce
Oname: 南台工商專科學校

[Organization-Club (Oname)]

台大佳韻音樂社 Oname: 佳韻音樂社
台大鋼琴社 Oname: 鋼琴社
杏林合唱團 Oname: 杏林合唱團
杏林弦樂團 Oname: 弦樂團
基克工作室 Oname: 基克工作室

[Organization-Goverment (Oname)]

網路博覽會 中華民國館 / Pavilion of Taiwan, R.O.C. Oname: 中華民國館
<A HREF="http://expo96.org.tw/Welcome_c.html">中華民國館</A> Oname: 中華民國館
<A HREF="http://www.motc.gov.tw/Welcome_c.html">交通館</A> Oname: 交通館
<A HREF="http://www.nmns.edu.tw/">國 立 自 然 科 學 博 物 館 </A> Oname: 國立自然科學博物館
台 北 市 立 動 物 園 Oname: 台北市立動物園
<A HREF="http://192.192.14.202/welcome.htm">國立中正文化中心</A> Oname: 國立中正文化中心
<A HREF="http://crab.ccl.itri.org.tw/cgi/m_normal">國家圖書館遠距圖書服務系統</A> Oname: 國家圖書館

[Personal Name (Pname)]

王立三的 HomePage / Li-San Wang's Homepage Pname: 王立三
王家俊 / John's House Pname: 王家俊
生命的照顧 － 范守仁醫師 / Life Care - Fan's Home Pname: 范守仁
何子之網頁 Pname: 何子
杜立群 Pname: 杜立群
邊國維的網頁 Pname: 邊國維
吳俊興 Pname: 吳俊興
吳振漢的窩 / Wilfred's HomePage Pname: 吳振漢
林育德（ AirL)的遊園地 Pname: 林育德
林欣蔚 / CELHW Pname: 林欣蔚
林信成的 W3 小棧 Pname: 林信成
依客那米克斯傳說—勇者耀耀之章 Pname: 那米克斯
阿哲的夢幻天地 Pname: 阿哲
林錦鴻 - 電腦玩家，網路流民，婦產科醫師 Pname: 林錦鴻
唐唐的世界 Pname: 唐唐
張正宜-不來不可的好地方 / TOM's Home Pname: 張正宜
黃兆談 Pname: 黃兆談
魚兒的小鎮－林康捷的 Homepage Pname: 林康捷
陳紀光 / HomePage of Chen Chi-kuang Pname: 陳紀光
陳炳宇 / Robin's Workgroup Pname: 陳炳宇
郭昇彥的烘焙機 Pname: 郭昇彥

Appendix C. Some Mapping Results in Content Part

In the following, Oname and Pname denote the extracted organization names and personal names respectively. The number indicates the token no. of the information in Web pages.

[Some Extracted Data in Content Sets before Mapping]

Oname: 資訊新館 63
E-Mail: root@csman.csie.ntu.edu.tw 59

Oname: 土木館 81
E-Mail: root@ce.ntu.edu.tw 82

Pname: 蔡博文 108
Oname: 地理系館 109
E-Mail: tsaibw@ccms.ntu.edu.tw 112

Pname: 丘台生 122
Oname: 漁科館 123
E-Mail: tschiu@ccms.ntu.edu.tw 124

Pname: 陳膺州 146
E-Mail: ingchen@chem60.ch.ntu.edu.tw 152

Pname: 黃靜美 171
E-Mail: mei@ccms.ntu.edu.tw 175

Pname: 林翰彥 178
Oname: 森林館 179
E-Mail: wenliang@ccms.ntu.edu.tw 180

Pname: 張震東 155
E-Mail: gdchang@ccms.ntu.edu.tw 160

Pname: 蘇明道 184
Oname: 農工館 185
E-Mail: sumd@ccms.ntu.edu.tw 186

Pname: 王友俊 382
E-Mail: wangecaa@ccms.ntu.edu.tw 387

Pname: 周伯戡 389
E-Mail: pkchou@ccms.ntu.edu.tw 391

Pname: 游張松 250
E-Mail: yucs@ccms.ntu.edu.tw 254

Pname: 曾珀雯 270
Pname: 徐信權 272
E-Mail: popo@ccms.ntu.edu.tw 276
E-Mail: kevins@ccms.ntu.edu.tw 277

[Some Mapping Results in Content Sets]

E-Mail: root@csman.csie.ntu.edu.tw	Oname: 資訊新館
E-Mail: focus@www.ntu.edu.tw	Oname: 焦點新聞
E-Mail: news@www.ntu.edu.tw	Oname: 網路新聞
E-Mail: campus@www.ntu.edu.tw	Oname: 校園新聞
E-Mail: tsaibw@ccms.ntu.edu.tw	Pname: 蔡博文
E-Mail: tschiu@ccms.ntu.edu.tw	Pname: 丘台生
E-Mail: ingchen@chem60.ch.ntu.edu.tw	Pname: 陳膺州
E-Mail: yucs@ccms.ntu.edu.tw	Pname: 游張松
E-Mail: hlee@cc.ntu.edu.tw	Pname: 李賢輝
E-Mail: popo@ccms.ntu.edu.tw	Pname: 曾珀雯
E-Mail: kevins@ccms.ntu.edu.tw	Pname: 徐信權
http: http://www.ntu.edu.tw/forest/R17.html	Oname: 國立臺灣大學森林學系暨研究所

Unknown Word Detection for Chinese by a Corpus-based Learning Method

Keh-Jiann Chen, Ming-Hong Bai
Institute of Information Science
Academia Sinica
Taipei, Taiwan
e-mail: kchen@iis.sinica.edu.tw, evan@iis.sinica.edu.tw

Abstract

One of the most prominent problems in computer processing of Chinese language is identification of the words in a sentence. Since there are no blanks to mark word boundaries, identifying words is difficult because of segmentation ambiguities and occurrences of out-of-vocabulary words (i.e. unknown words). In this paper, a corpus-based learning method is proposed which derives sets of syntactic rules that are applied to distinguish monosyllabic words from monosyllabic morphemes which may be parts of unknown words or typographical errors. The corpus-based learning approach has the advantages of 1. automatic rule learning, 2. automatic evaluation of the performance of each rule, and 3. balancing of recall and precision rates through dynamic rule set selection. The experimental results show that the rule set derived by the proposed method outperformed hand-crafted rules produced by human experts in detecting unknown words.

1. Introduction

One of the most prominent problems in computer processing of Chinese language is the identification of the words in a sentence. There are no blanks to mark the word boundaries in Chinese text. As a result, identifying words is difficult, because of segmentation ambiguities and occurrences of out-of-vocabulary words (i.e. unknown words). However most of the papers dealing with the problem of word segmentation focus their attention only on the resolution of ambiguous segmentation. The problem of unknown word identification is considered to be more difficult and needs to be

further investigated. Unknown words cause segmentation errors, because out-of-vocabulary words in an input text normally would be incorrectly segmented into pieces of single character word or shorter words. It is difficult to know when an unknown word is encountered since all Chinese characters can either be a morpheme or a word and there are no blanks to mark the word boundaries. Therefore without (or even with) syntactic or semantic checking, it is difficult to tell whether a character in a particular context is a part of an unknown word or whether it stands alone as a word. Compound words and proper names are the two major types of unknown words. There are many different types of compounds, such as nominal compounds, verbal compounds, determiner-measure compounds, numbers, reduplications etc. It is neither possible to list all of the compounds in the lexicon nor possible to write simple rules which can enumerate the compounds without over-generation or under-generation. Each different type of compound must be identified by either content or context dependent rules. Proper names and their abbreviations have less content regularity. Identifying them relies more on contextual information. The occurrence of typographical errors makes the problem even more complicated. There is currently no satisfactory algorithm for identifying both unknown words and typographical errors, but researchers are separately working on each different type of problem. Chang etc.[Chang etc. 94] used statistical methods to identify personal names in Chinese text which achieved a recall rate of 80% and a precision rate of 90%. Similar experiments were reported in [Sun etc. 94]. Their recall rate was 99.77%, but with a lower precision of 70.06%. Both papers deal with the recognition of Chinese personal names only. Chen & Lee [Chen & Lee 94] used morphological rules and contextual information to identify the names of organizations. Since organizational names are much more irregular than personal names in Chinese, they achieved a recall rate of 54.50% and a precision rate of 61.79%. A pilot study on automatic correction of Chinese spelling errors was done by Chang [Chang 94]. They used mutual information between a character and its neighboring words to detect spelling errors and then to automatically make the necessary corrections. The error detection process achieved a recall rate of 76.64% and a precision rate of 51.72%. Lin etc. [Lin etc. 93] made a preliminary study of the problem of unknown word identification. They used 17 morphological rules to recognize regular compounds and a statistical model to deal with irregular unknown words, such as proper names etc.. With this unknown word

resolution procedure, an error reduction rate of 78.34% was obtained for the word segmentation process. Since there is no standard reference data, the claimed accuracy rates of different papers vary due to different segmentation standards. In this paper we use the Sinica corpus as a standard reference data. The Sinica corpus is a word-segmented corpus based on the Chinese word segmentation standard for information processing proposed by ROCLING [Huang etc. 96, Chen etc. 96]. Therefore it contains many occurrences of unknown words which are separated by the blanks. The corpus were utilized for the purposes of training and testing. For the unknown word and typographical error identification, the following two steps are proposed. The first step is to detect the existence of unknown words and typographical errors. The second step is the recognition process, which determines the type and boundaries of each unknown word. The reasons for separating the detection process from the recognition process are as follows:

a. For different types of unknown words and typographical errors, they may share the same detection process, but have different recognition processes.
b. If the common method for spell checking is followed, the unknown word would be detected first, and a search for the best matching words would be performed next. Recognizing a Chinese word is somewhat different from spell checking, but they have a lot in common.
c. If the detection process performs well, the recognition process is better focused, making the total performance more efficient.

This paper focuses on the unknown word detection problem only (note that the typographical errors are considered as a special kind of unknown words). The problems of unknown word identification and typographical error correction will be left for future research. The unknown word detection problem and the dictionary-word detection problem are complementary problem, since if all known words in the input text can be detected, then the rest of character string would be unknown words. However this is not a simple task, since there are no blanks to delimit known words from unknown words. Therefore, the word segmentation process is applied first, and known words are delimited by blanks. Since unknown words are not listed in the dictionary, they will be segmented into shorter character/word sequences after a conventional dictionary-look-up word segmentation process. Sentence(1.b) shows the result of the word segmentation process on (1.a).

(1) a. 筑波大學延請七三年諾貝爾物理學獎得主江崎出任校長，

b. 筑 波 大學 延請 七三年 諾貝爾 物理學 獎 得主 江 崎 出任 校長，

According to an examination of a testing data which is a part of Sinica corpus, there are 4572 occurrences out of 4632 unknowns which were incorrectly segmented into sequence of shorter words and each sequence contains at least one monosyllabic word. That is, 60 of the unknown words were segmented into sequences of multi-syllabic words only. Therefore, the occurrences of monosyllabic words (i.e. single character words) in the segmented input text may denote the possible existence of unknown words. This is reasonable, since it is very rare that compounds or proper names are composed by several multi-syllabic words. Therefore the processes of detecting unknown words is equivalent to making the distinction between monosyllabic words and monosyllabic morphemes which are part of unknown words. Hence the complementary problem of unknown word detection is the problem of monosyllabic known-word detection. If all of the occurrences of monosyllabic words are considered as possible morphemes of unknown words was performed, the precision of the prediction is very low. When the word segmentation process on the Sinica corpus by a conventional dictionary look-up method, 69733 occurrences of monosyllabic words were found, but only 9343 were part of unknown words, a precision of 13.40%. In order to improve the precision, the monosyllabic words which properly fit in the contextual environment should be identified and should not be considered as possible morphemes of unknown words. In the next section, the corpus-based learning approach to identify contextually-proper monosyllabic words is introduced. In section 3, the experimental results are presented which includes a performance comparison between a hand-crafted method and the proposed corpus-based learning method.

2. Corpus-based Rule Learning for Identifying Monosyllabic Words

The procedure for detecting unknown words is roughly divided into three steps: 1. word segmentation, 2. part-of-speech tagging, 3. identification of contextually-proper monosyllabic words. The word segmentation procedure identifies words using a

dictionary look-up method and resolves segmentation ambiguities by maximizing the probability of a segmented word sequence[Chiang 92, Chang 91, Sproat 94] or by heuristic methods[Chen 92, Lee 91]. Either method can achieve very satisfactory results. Both have an accuracy of over 99%. For the purpose of unknown word identification, some regular types of compounds, such as numbers, determinant-measure compounds, and reduplication which have regular morphological structures, are also identified by their respective morphological rules during the word segmentation process[Chen 92, Lin 93]. The purpose of the second step, part-of-speech (pos) tagging, is for the convenience of step3 and the future process of unknown word identification. After pos tagging, sentence (1.b) becomes sentence (2); each word contains a unique pos.

(2) 筑(BOUND) 波(Nf) 大學(Nb) 延請(VC) 七三年(DM) 諾貝爾(Nb)
物理學(Na) 獎(Na) 得主(Na) 江(Na) 崎(BOUND) 出任(VG)
校長(Na)，.

Although the pos sequence may not be 100% correct, it is the most probable pos sequence in the terms of pos bi-gram statistics[Liu 95]. The details of the first two steps is not the major concerns of this paper. The focus is on the step of identifying contextually-proper monosyllabic words. Hereafter, for simplicity, the term 'proper-character' will denote a contextually-proper monosyllabic word and use the term 'improper-character' to denote a contextually-improper monosyllabic word which might be part of an unknown word. The way to identify proper-characters is by checking the following properties:

1. a proper-character should not be a bound-morpheme, and
2. the context of a proper-character should be grammatical.

Hence, if the character is a bound-morpheme, it will be considered possibly belonging to unknown word. However almost every character can function either as a word or as a bound morpheme. A character's functional role is contextually dependent. Therefore every monosyllabic word should be checked in its context for grammaticality by syntactic or semantic rules. For processing efficiency, such rules should be simple and have only local dependencies. It is not feasible to parse whole sentences in order to check whether or not characters are proper-characters. The task is then how to derive a set of rules which can be used to check the grammaticality of characters in context. If

the rules are too stringent, then too many proper-characters will be considered as improper-characters, resulting in a low precision rate. On the other hand if the rules are too relax, then too many improper-characters will be considered as proper-characters, resulting in a low recall rate. Therefore there is a tradeoff between recall and precision. In the case of unknown word detection, a higher recall rate and an acceptable precision rate is preferred. Writing hand-crafted rules is difficult, because there are more than 5ooo commonly used Chinese character and each of them may behave differently. A corpus-based learning approach is adapted to derive the set of contextual rules and to select the best set of rules by evaluating the performance of each individual rule. The approach is very similar to the error-driven learning method proposed by Brill [Brill 95].

Before the learning method is introduced, two commonly used measures for unknown word detection are defined. There are two types of unknown words. The type one unknown words contain monosyllabic morphemes. The type two unknown words are composed with multi-syllabic words only. Only the detection of the type one unknown word is considered here, since the occurrences of the type two unknown words are very rare as we mentioned before.

Recall Rate = # of unknown word detected / total number of unknowns

Precision Rate = # of correctly detected improper-characters / total # of guesses

An unknown word is considered successfully detected, if any one of its component is detected as an improper-character. It is noticed that the numerators for the recall rate and the precision rate are different, since if two (or more) components of an unknown word are detected as improper-characters, it is reasonable to count only one word detection but two improper-character detection. For the corpus-based learning method, a training corpus with all the words segmented and pos tagged is used. The monosyllabic words in the training corpus are instances of proper-characters and the words in the training corpus which are not in the dictionary are the instances of unknown words. Segmenting the unknown words by a dictionary look-up method produces the instances of improper-characters. By examining the instances of proper and improper characters and their contexts, the rule patterns and their performance evaluations can be derived and represent as a triplet (rule pattern, # of proper instances, # of improper instances). A contextual dependent rule may be:

a uni-gram pattern, such as '{的}', '{好}', '{(Nh)}', '{(T)}',

a bi-gram patterns, such as '{會}覺得', '{就}(VH)', '(Na){上}', '{(Dfa)}(Vh)', '(Ve){(Vj)}',

a tri-gram patterns, such as '{極}(VH)(T)', '(Na)(Dfa){高}',

where the string in the curly brackets will match a proper-character and the rest parts will match its context.

A good rule pattern has high applicability and high discrimination value (i.e. it occurs frequently and matches either proper-characters or improper-characters only, but not both). In fact no rule has perfect discriminating ability. Therefore a greedy method is adopted in selecting the best set of unknown word detection rules. A set of rules which can identify proper-characters with high accuracy is selected by sequentially choosing the rules which has the highest accuracy with applicability greater than a threshold value. The selected rule set is used as the recognition rules for proper-characters. The characters without a match by any one of the rules are considered as candidates of improper-character.

Rule selection algorithm:

1. Determine the threshold values for rule accuracy and applicability.

   For each rule Ri , when applied on the training corpus, the rule accuracy(Ri) = Mi / Ti, where Mi is the # of instances of matches of Ri with proper characters; Ti is the total # of matches of Ri. The rule applicability(Ri) = Ti.

2. Sequentially select the rules with the highest rule accuracy and the applicability greater than the threshold value, until there are no rules satisfying both threshold values.

The threshold value for rule accuracy controls the precision and recall performance of the final selected rule set. A higher accuracy requirement means less improper-characters would be wrongly recognized as proper-characters. Therefore the performance of such a rule set will have a higher recall value. However those proper-characters not matched with any rules will be mistaken as improper-characters which lowers precision. However on the other hand, if a lower accuracy threshold value is used, then most of the proper-characters will be recognized and many of the improper-characters will also be mistakenly recognized as proper-characters, resulting a lower recall rate and possibly a higher precision rate before reaching the maximal precision

value. Therefore if a detection rule set with a high recall rate is desired, the threshold value of rule accuracy must be set high. If precision is more important, then the threshold value must be properly adjusted lower to an optimal point. A balance between recall and precision should be considered.

In the next section, the experimental results on the different threshold values are presented. The threshold value for rule applicability controls the number of rules to be selected and ensures that only useful rules are selected.

The selected rule type may subsume another. Shorter rule patterns are usually more general than the longer rules. There are redundant rules in the initial rule selection. A further screening process is needed to remove the redundant rules. The screening process is based on the following fact: if a rule Ri is subsumed by rule Rj, then pattern of Ri is a sub-string of pattern Rj. For example the rule '{的}' is more general than the rule '{的} (Na)'.

Screening Algorithm:

a. Sort the rules according to their string patterns in increasing order, resulting in rules

   R1...Rn.

b. For i from 1 to n,

   if there is a j such that j< i ,and Rj is a sub-string of Ri , then remove Ri.

3. Experimental Results

The corpus-based learning method for unknown word detection was tested on the Sinica corpus which is a balanced Chinese corpus with segmented words tagged with pos [Huang 95, Chen 96]. The Sinica corpus version 2.0 contains 3.5 million words. 3 million words were used as the training corpus and 0.15 million words for the testing corpus. The word entries in the CKIP lexicon were considered as the known words. The CKIP lexicon contains about 80,000 entries of Chinese words with their syntactic categories and grammatical information[CKIP 93]. A word is considered as an unknown word, if it is not in the CKIP lexicon and not identified by the word segmentation program as a foreign word (for instance English,) a number, or a reduplicated compound. There were 53328 unknown words in the training corpus and

4632 unknown words in the testing corpus. A few bi-word compounds were deliberately ignored as unknowns, such as '分析化學 analytical chemistry ','技術人員 technical member'...,since they are not identifiable by any algorithm which does not incorporate real world knowledge. In addition, whether these are single compounds or noun phrases made up of two words is debatable. In fact ignoring the bi-word compounds did not affect the results too much, since the fact that there were only 60 such unknown words out of 4632 shows that they rarely occurred in the corpus.

The following types of rule patterns were generated from the training corpus. Each rule contains a token within curly brackets and its contextual tokens without brackets. For some rules there may be no contextual dependencies.

Rule type	Examples
char	{的}
word char	不 {願}
char word	{全} 世界
category	{(T)}
{category} category	{(Dfa)} (Vh)
category {category}	(Na) {(Vcl)}
char category	{就} (VH)
category char	(Na) {上}
category category char	(Na) (Dfa) {高}
char category category	{極} (Vh) (T)

Rules of the 10 different types of patterns above were generated automatically by extracting each instance of monosyllabic words in the training corpus. Every generated rule pattern was checked for redundancy and the frequencies of proper and improper occurrences were tallied. For instance, the pattern '{的}' occurred 165980 times in the training corpus; 165916 of these were proper instances and 64 of these were improper instances (i.e. 64 times "的" occurred as part of an unknown word). Appendix 1 shows some of the rule patterns and their total occurrences counts as well as the number of improper instances. At the initial stage, 1455633 rules were found. After eliminating the rules with frequency less than 3, 215817 rules remained. At next

stage different rule selection threshold values were used to generate 10 different sets of rules. These rule sets were used to detect unknown words in the testing corpus. The testing corpus contained 152560 words. In the first step, the running text of the testing corpus was segmented into words by a dictionary look-up method and then tagged with their part-of-speech by an automatic tagging process. Each different rule set was applied to detect the unknown words in the testing corpus. The characters without a match will be considered as part of an unknown word. The performance results of different rule sets are shown in Table 2 and the detail statistics are shown in Appendix 3.

The results show that there is a tradeoff between precision and recall rate, but the overall performance was much better than when hand-crafted rules written by human experts were used. The set of hand-crafted rules were written by linguists. They examined the training corpus and wrote up the rule set for proper-characters to the best of their ability. The hand-craft rules had a precision rate of 39.11% and a recall rate of 81.45% which are much lower than the rule set made by the corpus-based rule learning method. The syntactic complexity of monosyllabic words was the reason for the lower coverage of the hand-crafted rules. There were only 139 hand-crafted rules while the proposed method generated thousands of rules as shown in Table 2. The number of rule selected is increasing with respect to the decrement of the accuracy of rule selection criteria, because more rules will satisfy the lower accuracy requirement. However the number of rules after the screening process is decreasing in accordance with the decrement of the accuracy of the rule selection criteria. For instances there are 207059 number and 210552 number of rules selected respectively for the rule accuracy criterion of 98% and 95%, but after the screening process the number of rules become 70415 and 56020. The reason for this interesting fact is that to achieve a higher accuracy demands more contextual dependency rules to discriminate between proper-characters and improper-characters; on the other hand lower accuracy requirement may cause the inclusion of more shorter rules which eliminate a lot of longer rules subsumed by the shorter rules.

Rule selection criteria	Recall rate	Precision rate	# of rules after screening
(0) no rule applied	100%	13.40%	0

(1)	rule accuracy >= 55%	63.32%	73.69%	12996
(2)	rule accuracy >= 60%	63.89%	73.73%	15250
(3)	rule accuracy >= 65%	64.85%	74.04%	17875
(4)	rule accuracy >= 70%	68.18%	74.61%	18559
(5)	rule accuracy >= 75%	73.80%	74.36%	20191
(6)	rule accuracy >= 80%	77.34%	73.26%	23047
(7)	rule accuracy >= 85%	81.06%	71.52%	30097
(8)	rule accuracy >= 90%	87.40%	68.74%	36563
(9)	rule accuracy >= 95%	93.66%	64.73%	56020
(10)	rule accuracy >= 98%	96.30%	60.62%	70415

Note: all of the applicability values are set to rule frequency >= 3.

==========

Table 2. The experimental results of unknown word detection on the testing corpus

4. Conclusion and Future Research

The corpus-based learning approach proved to be an effective and easy method of finding the unknown word detection rules. The advantages of using a corpus-based method are as follows:

a. The syntactic patterns of proper-characters are complicated and numerous. It is hard to hand-code each different patterns, yet most high frequency patterns are extractable fromthe corpus.

b. The corpus provides a standard reference data not only for rule generation but also for rule evaluation. The hand-craft rules can also be evaluated automatically and be incorporated into the final detection rule set, if the rule has a high accuracy rate.

c. It is easy to control the balance between the precision and the recall of the detection algorithm, since we know the performance of each detection rule based on the training corpus.

Different types of unknown words have different levels of difficulties in identifying them. The detection of compounds is the most difficult because some of their morphological structures are similar to common syntactic structures. The detection of proper names and typographical errors are believed to be easier because

of their irregular syntactic patterns. The results with respect to different types of syntactic categories were checked. Appendix 3 shows that the recall rates of proper names (i.e. category Nb), is less affected by the higher precision requirement. there was no data for typos, but the detection of typos is believed to similar to the detection of proper names; that is, a higher precision can be achieved without sacrificing the recall rate. If a parallel corpora with and without typos is available, the corpus-based rule learning method could also be applied to the detection of typographical errors in Chinese.

After the unknown word detection process, an identification algorithm will be required to find the exact boundaries and the part-of-speech of each unknown word. This will require future research. Different types of rules will be required in identifying different compounds and proper names. The corpus can still play an essential role in the generation of the rules and their evaluation.

Acknowledgments

The authors wish to thank Dr. Charles Lee and the anonymous reviewers for their useful comments on this paper.

References

Brill, Eric, 1995," Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging," *Computational Linguistics* Vol.21, No. 4, pp.543-566.

Chang, J. S., C. D. Chen, & S. D. Chen, 1991," Word Segmentation through Constraint Satisfaction and Statistical Optimization," Proceedings of ROCLING IV, pp. 147-165.

Chang, C. H., 1994,"A Pilot Study on Automatic Chinese Spelling Error Correction" *Communication of COLIPS*, Vol.4 No. 2, 143-149.

Chang J. S.,S.D. Chen, S. J. Ker, Y. Chen, & J. Liu,1994 "A Multiple-Corpus Approach to Recognition of Proper Names in Chinese Texts", *Computer Processing of Chinese and Oriental Languages*, Vol. 8, No. 1, 75-85.

Chen, H.H., & J.C. Lee, 1994,"The Identification of Organization Names in Chinese Texts", *Communication of COLIPS*, Vol.4 No. 2, 131-142.

Chen, K.J., C.R. Huang, L. P. Chang & H.L. Hsu, 1996,"SINICA CORPUS: Design Methodology for Balanced Corpora," *Proceedings of PACLIC 11th Conference*, pp.167-176.

Chen, K.J. & S.H. Liu, 1992,"Word Identification for Mandarin Chinese Sentences," *Proceedings of 14th Coling*, pp. 101-107.

Chiang, T. H., M. Y. Lin, & K. Y. Su, 1992," Statistical Models for Word Segmentation and Unknown Word Resolution," Proceedings of ROCLING V, pp. 121-146.

Huang, C. R. Et al.,1995,"The Introduction of Sinica Corpus," *Proceedings of ROCLING VIII*, pp. 81-89.

Lee,H.J. & C.L. Yeh, 1991, "Rule-based Word Identification for Mandarin Chinese Sentences- A Unification Approach," *Computer Processing of Chinese and Oriental Languages*, Vol. 5, No. 1, 97-118.

Lin, M. Y., T. H. Chiang, & K. Y. Su, 1993," A Preliminary Study on Unknown Word Problem in Chinese Word Segmentation," Proceedings of ROCLING VI, pp. 119-137.

Liu S. H., K. J. Chen, L.P. Chang, & Y.H. Chin, 1995, "Automatic Part-of-Speech Tagging for Chinese Corpora," *Computer Processing of Chinese and Oriental Languages*, Vol. 9, No. 1, 31-48.

Sproat, R., C. Shih, W. Gale, & N. Chang,1994, "A Statistical Finite-State Word-Segmentation Algorithm for Chinese," *Proceedings of 32nd ACL Conference*.

Sun, M. S., C.N. Huang, H.Y. Gao, & Jie Fang, 1994, "Identifying Chinese Names in Unrestricted Texts", *Communication of COLIPS*, Vol.4 No. 2, 113-122.

Appendix 1. Samples of rule patterns

rule	frequency	error	accuracy
{的}	165980	64	99.71 %
{是}	41089	78	98.10 %
{也}	16066	11	99.31 %
{她}	6185	4	99.35 %
{這}	5046	1	99.80 %
{或}	4582	3	99.34 %
{該}	2302	2	99.13 %
{(T)}	177641	177	99.00 %
{(Nh)}	73034	344	99.53 %
{(Caa)}	46659	392	99.16 %
{(SHI)}	41089	78	99.81%
{(Dfa)}(VH)	11037	39	99.65 %
{(Nh)}(Na)	6640	62	99.07 %
{(P)}(Nh)	6247	52	99.17 %
{(Nep)}(Na)	4030	26	99.35 %
(Na){(VCL)}	8062	299	96.30 %
(VC){(Di)}	4155	76	98.18 %
(VE){(VJ)}	1884	46	97.56 %
(VJ){(VJ)}	1489	53	96.44 %
(VJ){(Dfa)}	1004	5	99.50 %
{與}(Na)	3933	6	99.85 %
{及}(Na)	2831	18	99.36 %
{在}(VC)	2451	2	99.92 %
(VH){地}	1787	14	99.22 %
(VC){者}	1731	1	99.94 %
(Na){很}	1172	0	100 %
{再}(VC)(Na)	221	0	100 %
{令}(Na)(VH)	200	0	100 %
{各}(Na)(Na)	190	3	98.42 %
{極}(VH)(T)	187	1	99.47 %
(Na)(Dfa){高}	263	0	100 %
(Na)(VH){地}	248	1	99.60 %
(Na)(Na){時}	231	2	99.14 %
(T)(Na){則}	174	0	100 %
{會}覺得	139	1	99.28 %
{才}知道	124	0	100 %
{拿}著	121	0	100 %
{迄}今	117	0	100 %
的{話}	1406	2	99.86 %
並{非}	319	0	100 %

Appendix 2. Samples of testing results

First line contains the original text. The second line shows the result of word segmentation and pos tagging. The third line is the result of unknown word detection such that the improper-characters are marked with '(?)'.

****

有的時候我想吃點美國菜,
有的(Neqa) 時候(Na) 我(Nh) 想(VE) 吃(V) 點(Na) 美國(Nc) 菜(Na),
有的()(Neqa) 時候()(Na) 我()(Nh) 想()(VE) 吃()(V) 點()(Na) 美國()(Nc) 菜(?)(Na),

****

微軟過去兩年也推出了近百種新產品，
微(D) 軟(VH) 過去(Nd) 兩年(DM) 也(D) 推出(VC) 了(VJ) 近百種(DM) 新(VH) 產品(Na)，
微()(D) 軟(?)(VH) 過去()(Nd) 兩年()(DM) 也()(D) 推出()(VC) 了()(VJ) 近百種()(DM) 新()(VH) 產品()(Na)，

****

即使營收和獲利成長開始減慢，
即使(Cbb) 營收(Na) 和(Caa) 獲利(VH) 成長(VH) 開始(VL) 減(VJ) 慢(VH)，
即使()(Cbb) 營收()(Na) 和()(Caa) 獲利()(VH) 成長()(VH) 開始()(VL) 減(?)(VJ) 慢(?)(VH)，

****

一九九四將是日本教育的改革年，
一九九四(DM) 將(D) 是(SHI) 日本(Nc) 教育(VC) 的(T) 改革(VC) 年(Nf)，
一九九四()(DM) 將()(D) 是()(SHI) 日本()(Nc) 教育()(VC) 的()(T) 改革()(VC) 年(?)(Nf)，

****

日本可能出現第一個個人主義世代。
日本(Nc) 可能(D) 出現(VH) 第一個(DM) 個(Nf) 人(Na) 主義(Na) 世代(Na)。
日本()(Nc) 可能()(D) 出現()(VH) 第一個()(DM) 個(?)(Nf) 人(?)(Na) 主義()(Na) 世代()(Na)。

****

筑波大學延請七三年諾貝爾物理學獎得主江崎出任校長，
筑(BOUND) 波(Nf) 大學(Nb) 延請(VC) 七三年(DM) 諾貝爾(Nb) 物理學(Na) 獎(Na) 得主(Na) 江(Na) 崎(BOUND) 出任(VG) 校長(Na)，
筑(?)(BOUND) 波(?)(Nf) 大學()(Nb) 延請()(VC) 七三年()(DM) 諾貝爾()(Nb) 物理學()(Na) 獎(?)(Na) 得主()(Na) 江(?)(Na) 崎(?)(BOUND) 出任()(VG) 校長()(Na)，

****

就連整個體系中最官僚的教育當局－日本文部省，
就(Da) 連(D) 整個(DM) 體系(Na) 中(Ng) 最(Dfa) 官僚(Na) 的(T) 教育(VC) 當局(Na) －(BOUND) 日本(Nc) 文(BOUND) 部(Nc) 省(Nc),
就()(Da) 連()(D) 整個()(DM) 體系()(Na) 中()(Ng) 最()(Dfa) 官僚()(Na) 的()(T) 教育()(VC) 當局()(Na) －()(BOUND) 日本()(Nc) 文(?)(BOUND) 部(?)(Nc) 省(?)(Nc)，

****

也在調整一向溫吞的改革步伐。
也(D) 在(VCL) 調整(VC) 一向(D) 溫(VHC) 吞(VC) 的(T) 改革(VC) 步伐(Na)。
也()(D) 在()(VCL) 調整()(VC) 一向()(D) 溫(?)(VHC) 吞(?)(VC) 的()(T) 改革()(VC) 步伐()(Na)。

****

業者可以更準確地捕捉各個特定人口群，
業者(Na) 可以(D) 更(D) 準確(VH) 地(Na) 捕捉(VC) 各個(DM) 特定(A) 人口(Na) 群(Nf)，
業者()(Na) 可以()(D) 更()(D) 準確()(VH) 地()(Na) 捕捉()(VC) 各個()(DM) 特定()(A) 人口()(Na) 群(?)(Nf)，

****

Appendix 3. The detail performance results of the different rule sets

The first column shows the categories of unknown words.
The second column is the number of occurrences of the unknown words with the category shown in column one.
The third column is the number of unknown words detected.

The last column is the recall rate. Category	# of unknown words	Frequency > 2						
		Accuracy >=						
		55%	60%	70%	80%	90%	95%	98%
A	63	66.67%	66.67%	66.67%	74.60%	79.37%	87.30%	96.83%
Na	1396	75.07%	76.29%	79.87%	85.24%	92.12%	95.85%	97.13%
Nb	1511	87.16%	87.56%	90.47%	95.90%	98.28%	99.47%	99.60%
Nc	424	67.92%	67.92%	74.76%	75.94%	89.86%	91.04%	95.52%
Nd	24	16.67%	16.67%	25.00%	37.50%	50.00%	79.17%	83.33%
Nh	62	4.84%	4.89%	35.48%	75.81%	88.71%	90.32%	93.55%
VA	151	31.79%	32.45%	34.44%	54.30%	69.54%	83.44%	86.76%
VB	25	20.00%	20.00%	24.00%	40.00%	64.00%	84.00%	84.00%
VC	439	14.58%	14.58%	20.05%	41.91%	73.13%	89.29%	94.99%
VCL	63	14.29%	14.29%	15.87%	36.51%	79.37%	90.48%	96.83%
VD	48	2.08%	2.08%	8.33%	56.25%	77.08%	89.58%	93.75%
VE	70	4.29%	4.29%	4.29%	12.86%	24.29%	78.57%	88.57%
VG	69	7.25%	7.25%	10.15%	21.74%	40.58%	69.57%	86.96%
VH	137	22.65%	24.09%	35.77%	60.58%	73.72%	84.67%	89.78%
VHC	23	91.30%	91.30%	91.30%	95.65%	95.65%	95.65%	95.65%
VJ	67	8.96%	8.96%	11.94%	25.37%	44.78%	67.16%	83.58%
Total:	4572							
Recall:		63.32%	63.89%	68.18%	77.34%	87.40%	93.66%	96.30%
Precision:		73.70%	73.73%	74.61%	73.27%	68.74%	64.73%	60.63%

Analyzing the Complexity of a Domain With Respect To An Information Extraction Task

Amit Bagga*
Alan W. Biermann
Dept. of Computer Science
Box 90129, Duke University
Durham, N. C. 27708–0129. USA
Internet: {amit, awb}@cs.duke.edu

Abstract

In this paper we describe a method of classifying facts (information) into categories or levels; where each level signifies a different degree of difficulty of extracting the fact from a piece of text containing it. Based on this classification mechanism, we also propose a method of evaluating a domain by assigning to it a "domain number" based on the levels of a set of *standard* facts present in the articles of that domain. In addition, we undertake two studies. The first study evaluates the effect of levels on the performance of message understanding systems while the second evaluates the effect of discourse processing, specifically coreferencing, on the performance of message understanding systems.

*Supported by a fellowship from IBM Corporation.

1 Introduction

The Message Understanding Conferences (MUCs) have been held with the goal of qualitatively evaluating message understanding systems. The six MUCs held thus far have been quite successful at providing such an evaluation. Since MUC-3, the systems have been evaluated on three different domains, and the task has been expanded from simply filling templates, in MUC-3 (MUC-3, 1991), to including named entity recognition (NE) and coreferencing (CO), in MUC-6 (MUC-6, 1995), as well. For MUC-6, the precision statistics of the participating systems varied from 34% to 73% and the recall statistics varied from 32% to 58% on the scenario template (ST) task.

But while the MUCs have shown the differences in the performance of the systems for a particular task (in a particular domain), little or no work has been done in trying to explain the differences in the performance of the systems. In addition, very little work has been done in analyzing the difficulty of understanding a text in a particular domain; both, independently, as well as in comparison to understanding a text in some other domain.

The organizers of MUC-5 attempted to compare the difficulty of the EJV (English Joint Ventures) task in MUC-5 to the terrorist task of MUC-3 and MUC-4. The criteria used for comparing these two tasks included the vocabulary size, the average sentence length, the average number of sentences per text, the number of texts, etc. (Sundheim, 1993). The organizers of MUC-6 did not attempt to compare the difficulty of the MUC-6 task to the previous MUC tasks saying that "the problem of coming up with a reasonable, objective way of measuring relative task difficulty has not been adequately addressed" (Sundheim, 1995).

In this paper we describe a method of classifying facts (information) into categories or levels; where each level signifies a different degree of difficulty of extracting the fact from a piece of text containing it. Moreover, we also propose a method of evaluating a domain by assigning to it a "domain number" based on the levels of a set of *standard* facts present in the articles of that domain. Based on our classification mechanism, we undertake two studies. The first one evaluates the the performance of three MUC systems (BBN, NYU, and SRI) based on their ability to extract a set of "standard" facts (at different levels) from the MUC-4 terrorist reports domain. The second study evaluates the effect of discourse processing, specifically coreferencing, on the performance of the three systems.

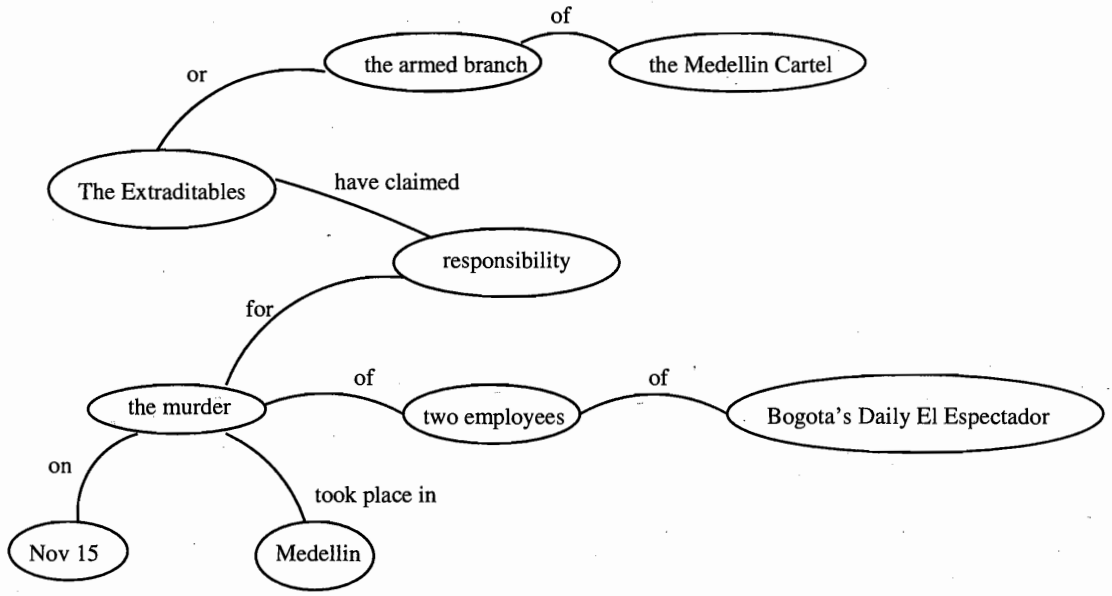


Figure 1: A Sample Semantic Network

2 Definitions

Semantic Network:

A *semantic network* consists of a collection of nodes interconnected by an accompanying set of arcs. Each node denotes an object and each arc represents a binary relation between the objects. (Hendrix, 1979)

A Partial Semantic Network:

A *partial semantic network* is a collection of nodes interconnected by an accompanying set of arcs where the collection of nodes is a subset of a collection of nodes forming a semantic network, and the accompanying set of arcs is a subset of the set of arcs accompanying the set of nodes which form the semantic network.

Figure 1 shows a sample semantic network for the following piece of text:

> "The Extraditables," or the Armed Branch of the Medellin Cartel have claimed responsibility for the murder of two employees of Bogota's daily El Espectador on Nov 15. The murders took place in Medellin.

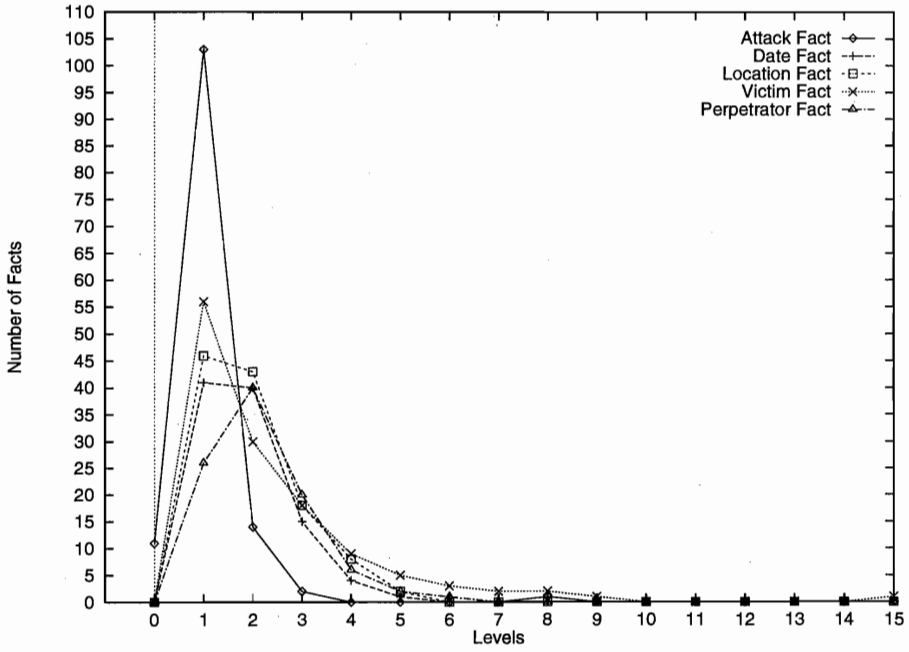


Figure 2: MUC-4: Level Distribution of Each of the Five Facts

3 The Level of A Fact

The level of a fact, F, in a piece of text is defined by the following algorithm:

1. Build a semantic network, S, for the piece of text.

2. Suppose the fact, F, consists of several nodes $\{x_1, x_2, \ldots, x_n\}$. Let s be the partial semantic network consisting of the set of nodes $\{x_1, x_2, \ldots, x_n\}$ interconnected by the set of arcs $\{t_1, t_2, \ldots, t_k\}$.

   We define the *level* of the fact, F, *with respect to* the semantic network, S to be equal to k, the number of arcs linking the nodes which comprise the fact F.

3.1 Observations

Given the definition of the level of a fact, the following observations can be made:

- The level of a fact is related to the concept of "semantic vicinity" defined by Schubert et. al. (Schubert, 1979). The *semantic vicinity* of a node in a semantic net consists of the nodes and the arcs reachable from that node by traversing a small number of arcs. The fundamental assumption used here is that "the knowledge required to

perform an intellectual task generally lies in the semantic vicinity of the concepts involved in the task" (Schubert, 1979).

The level of a fact is equal to the number of arcs that one needs to traverse to reach all the concepts (nodes) which comprise the fact of interest.

- A level-0 fact consists of a single node (i.e. no transitions) in a semantic network.
- A level-k fact is a *union* of k level-1 facts.
- Conjunctions/disjunctions increase the level of a fact.
- The higher the level of a fact, the harder it is to extract it from a piece of text.
- A fact appearing at one level in a piece of text may appear at some other level in the same piece of text.
- The level of a fact in a piece of text depends on the granularity of the semantic network constructed for that piece of text. Therefore, the level of a fact with respect to a semantic network built at the word level (i.e. words represent objects and the relationships between the objects) will be greater than the level of a fact with respect to a semantic network built at the phrase level (i.e. noun groups represent objects while verb groups and preposition groups represent the relationships between the objects).

3.2 Examples

Let S be the semantic network shown in Figure 1. S has been built at the phrase level.

- The city mentioned, in S, is an example of a level-0 fact because the "city" fact consists only of one node "Medellin."
- The type of attack, in S, is an example of a level-1 fact.

  We define the *type of attack* in the semantic network to be an attack designator such as "murder," "bombing," or "assassination" with one modifier giving the victim, perpetrator, date, location, or other information.

  In this case the type of attack fact is composed of the "the murder" and the "two employees" nodes and their connector. This makes the type of attack a level-1 fact.

The type of attack could appear as a level-0 fact as in "the Medellin bombing" (assuming that the semantic network is built at the phrase level) because in this case both the attack designator (bombing) and the modifier (Medellin) occur in the same node. The type of attack fact occurs as a level-2 fact in the following sentence (once again assuming that the semantic network is built at the phrase level): "10 people were killed in the offensive which included several bombings." In this case there is no direct connector between the attack designator (several bombings) and its modifier (10 people). They are connected by the intermediatory "the offensive" node; thereby making the type of attack a level-2 fact. The type of attack can also appear at higher levels.

- In S, the date of the murder of the two employees is an example of a level-2 fact. This is because the attack designator (the murder) along with its modifier (two employees) account for one level and the arc to "Nov 15" accounts for the second level.

  The date of the attack, in this case, is not a level-1 fact (because of the two nodes "the murder" and "Nov 15") because the phrase "the murder on Nov 15" does not tell one that an attack actually took place. The article could have been talking about a seminar on murders that took place on Nov 15 and not about the murder of two employees which took place then.

- In S, the location of the murder of the two employees is an example of a level-2 fact. The exact same argument as the date of the murder of the two employees applies here.

- The complete information, in S, about the victims is an example of a level-2 fact because to know that two employees of Bogota's Daily El Espectador were victims, one has to know that they were murdered. The attack designator (the murder) with its modifier (two employees) accounts for one level, while the connector between "two employees" and "Bogota's Daily El Espectador" accounts for the other.

- Similarly, the complete information, in S, about the perpetrators of the murder of the two employees is an example of a level-5 fact. The breakup of the 5 levels is as follows: the fact that two employees were murdered accounts for one level; the fact

that "The Extraditables" have claimed responsibility for the murders accounts for two additional levels; and the fact that the Extraditables are the "armed branch of the Medellin Cartel" account for the remaining two levels.

4 Justification of the Methodology

The level of a fact quantifies the "spread" in the information that makes up the fact. Therefore, the higher the level of a fact, the greater is the "spread" in the information that makes up the fact. This means that more processing has to be done to identify and link all the individual pieces of information that make up the fact. In fact, an exploratory study done by Beth Sundheim during MUC-3 showed "a degradation in correctness of message processing as the information distribution in the message became more complex, that is, as slot fills were drawn from larger portions of the message and required more discourse processing to extract the information and reassemble it correctly in the required template(s)" (Hirschman, 1992).

An argument can be made that there are other factors, apart from the spread of information, which influence the difficulty of extracting a fact from text. Some of these factors include the amount of training done on an information extraction system, the quality of training, and the frequency of occurrence of the patterns that a system has been trained on. While these factors do influence the performance of an information extraction system and they do give some indication as to how difficult it was for a particular system to extract the fact, they do not give a system independent way of determining the complexity of extracting the fact.

In (Hirschman, 1992), Lynette Hirschman proposed the following hypothesis: there are facts that are simply harder to extract, across all systems. Based on our definition of the level of a fact, we analyzed the performances of three different information extraction systems on the MUC-4 terrorist reports domain. Our analysis shows that all the three systems consistently did much worse on higher level facts. In addition to confirming Hirschman's hypothesis, the analysis also shows that higher level facts are indeed harder to extract. Some details of the analysis are given later in this paper. (Bagga, 1997) gives the complete details about the analysis.

5 Building the Semantic Networks

As mentioned earlier, the level of a fact for a piece of text depends on the semantic network constructed for the text. Since there is no unique semantic network corresponding to a piece of text, care has to be taken so that the semantic networks are built consistently.

For the set of experiments described in the rest of the paper we used the following algorithm to build the semantic networks:

1. Every article was broken up into a non-overlapping sequence of noun groups (NGs), verb groups (VGs), and preposition groups (PGs). The rules employed to identify the NGs, VGs, and PGs were almost the same as the ones employed by SRI's FASTUS system[1].

2. The nodes of the semantic network consisted of the NGs while the transitions between the nodes consisted of the VGs and the PGs.

3. Identification of coreferent nodes and prepositional phrase attachments were done manually.

Obviously, if one were to employ a different algorithm for building the semantic networks, one would get different numbers for the level of a fact. But, if the algorithm were employed consistently across all the facts of interest and across all articles in a domain, the numbers on the level of a fact would be consistently different and one would still be able to analyze the relative complexity of extracting that fact from a piece of text in the domain.

6 Analysis of MUC-4

Based on our definition of the level of a fact, we analyzed the MUC-4 terrorist domain. Based on the official MUC-4 template, we selected a set of *standard* facts that we felt captured most of the information in the template. They are: (The full definition of each fact is not included here.)

- The type of attack.

[1]We wish to thank Jerry Hobbs of SRI for providing us with the rules of their partial parser.

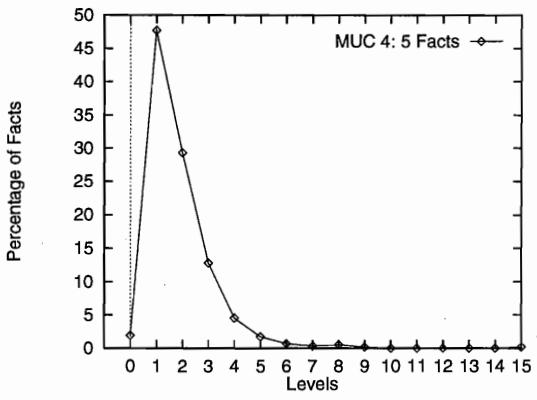


Figure 3: MUC-4: Level Distribution of the Five Facts Combined

- The date of the attack.
- The location of the attack.
- The victim (including damage to property).
- The perpetrator(s) (including suspects).

We then built the semantic networks (using the algorithm described in the previous section) for the relevant articles from the MUC-4 TST3 set of 100 articles. From the semantic network for each article, we calculated the levels of each of the five standard facts. The level distribution of the five facts for the MUC-4 TST3 set is shown in Figure 2. The level distribution of the five facts combined is shown in Figure 3.

Based on the data collected above, we made the following observations:

- There were 69 relevant articles in the MUC-4 TST3 set of 100 articles, each reporting one or more terrorist attacks.
- The five facts of interest appeared 570 times in the 69 articles.
- A number of articles reported the same fact at two different places and at two different levels in the same article. The first, usually, in the first paragraph of the text which reported the attack without giving too many details, and, the second, later in the article when the attack was reported with all the details.

  As one would expect, the level of the first occurrence of a fact in an article is usually less than or equal to the level of the second occurrence of that fact in the same article.

- From Figure 3, we can see that almost 50% of the five facts were at level-1. This is not surprising because four out of the five *standard* facts most frequently occur as level-1 facts (Figure 2).

6.1 Evaluating the Difficulty of the MUC-4 Terrorist Domain

We extended our analysis to analyze the difficulty of understanding a text in the MUC-4 terrorist domain.

Obviously, the difficulty of understanding a text in a domain depends directly on the expected level of a fact in that domain. We define this expected level of a fact in a domain to be the *domain number* of the domain. The domain number is measured in level units (LUs). Two domains can therefore be compared on the basis of their domain numbers.

The formula used to calculate the domain number is:

$$\frac{\sum_{l=0}^{\infty} l * x_l}{\sum_{l=0}^{\infty} x_l}$$

where x_l is the number of times one of the *standard* facts appeared at level-l in the articles of the domain.

Based on the levels of the five standard facts in the MUC-4 TST3 set of articles, we calculated the domain number of the terrorist domain to be 1.87 LUs. We are assuming the fact that the set of 100 randomly chosen articles in the MUC-4 TST3 set are representative of the domain. This assumption may not necessarily hold, but, given the large number of articles we analyzed, we hope that the domain number calculated is close to the actual domain number of the terrorist domain.

7 Analysis of MUC-5

Because two different domains were used in MUC-5 (each in two different languages), we decided to focus only on the English Joint Ventures (EJV) domain. Once again, the set of *standard* facts were selected from the official MUC-5 template and were chosen such that they contained most of the information in the template. They are: (The full definition of each fact is not included here.)

- The parent(s) of the joint venture formed.

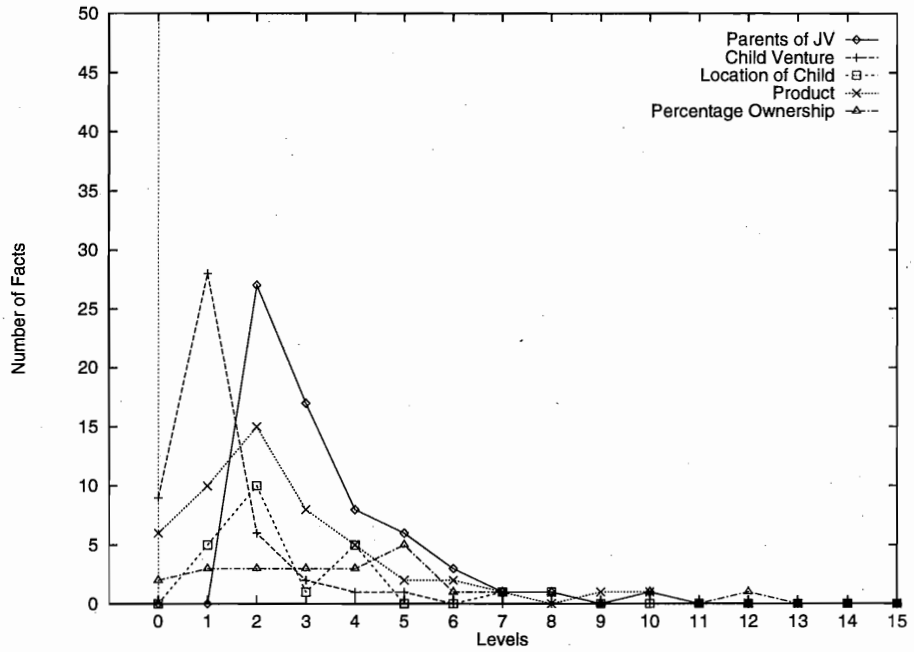


Figure 4: MUC-5: Level Distribution of Each of the Five Facts

- The child joint venture formed.
- The location of the child.
- Product that the child will produce.
- Percentage ownership of each parent.

Due to the unavailability of the official test set used for the MUC-5 EJV evaluation, we used a set of 50 articles used by the systems for training on the EJV domain. Using the algorithm described earlier, we then built the semantic networks for the relevant articles. Out of the 50 articles, 47 were relevant and the five *standard* facts appeared 209 times in these articles. The level distribution of each of the five facts is shown in Figure 4. The level distribution of the five facts combined is shown in Figure 5. Based on Figure 4 one can deduce that the MUC-5 EJV domain is harder than the MUC-4 terrorist domain because three out of the five standard facts most frequently occur as level-2 facts. Figure 5 peaks at level-2 giving further indication that the domain number for this domain is more than 2 LUs.

Based on the levels of the *standard* set of facts, we calculated the domain number of the MUC-5 EJV domain to be 2.67 LUs. This domain number is almost 1 LU higher than

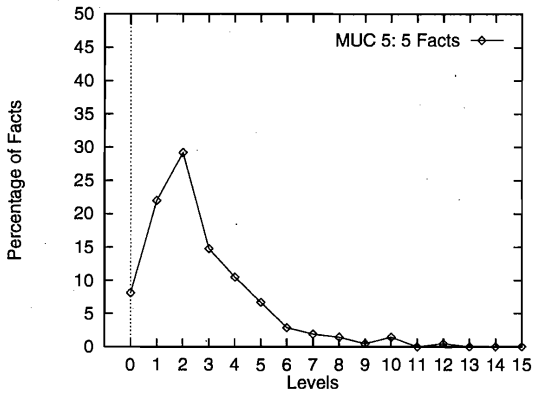


Figure 5: MUC-5: Level Distribution of the Five Facts Combined

the domain number for the MUC-4 terrorist attack domain and it shows that the MUC-5 EJV task was much harder than the MUC-4 task. In comparison, an analysis, using more "superficial" features, done by Beth Sundheim, shows that the nature of the MUC-5 EJV task is approximately twice as hard as the nature of the MUC-4 task (Sundheim, 1993).

8 Analysis of MUC-6

The domain used for MUC-6 consisted of articles regarding changes in corporate executive management personnel. As in the case of our analyses of the previous two MUCs, we selected a set of *standard* facts based on the official MUC-6 template. This set consisted of the following facts: (The full definition of each fact is not included here.)

- Organization where the change(s) in the personnel took place.
- The position involved in the changes.
- The person coming in to the position.
- The person leaving the position.
- The company/post from where the person coming in is hired.
- The company/post that the person going out is going to.

We analyzed the levels of the *standard* set of facts in the official MUC-6 test set by building the semantic networks for the relevant articles in the test set (using the algorithm

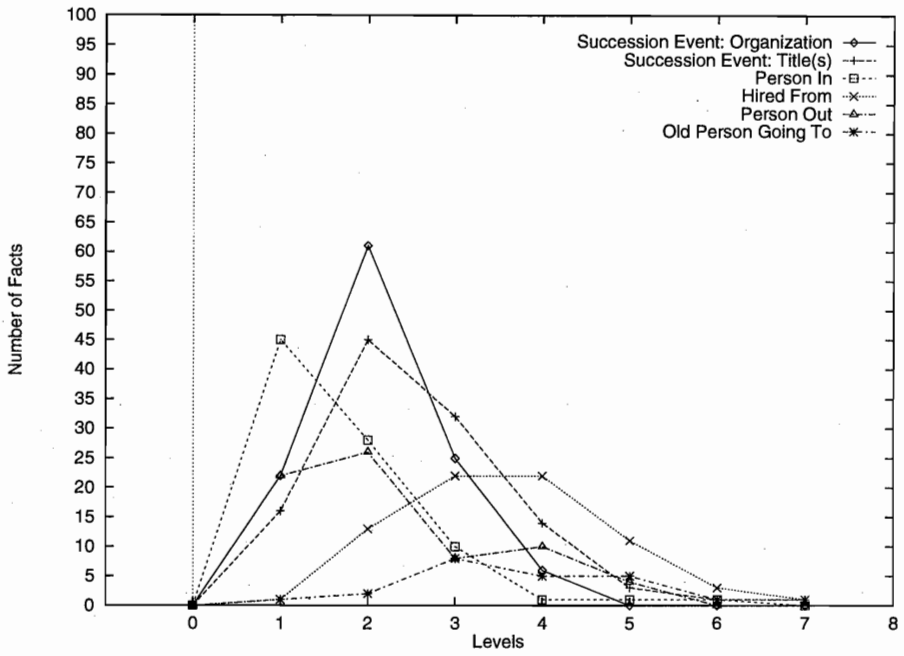


Figure 6: MUC-6: Level Distribution of Each of the Six Facts

described earlier). This test set consisted of 100 articles, 56 of which were relevant. The six *standard* facts appeared 478 times in the relevant articles. The level distribution of each of these six facts is shown in Figure 6. The level distribution of these six facts combined is shown in Figure 7.

We calculated the domain number for the MUC-6 domain to be 2.47 LUs. This indicates that the MUC-6 domain is almost as hard as the MUC-5 EJV domain. Figure 8 shows the domain numbers for the three MUCs that have been analyzed.

9 Extending the Analysis

Motivated by the exploratory study done by Beth Sundheim, we decided to undertake two studies. The first one was to do an analysis regarding the levels of facts (the distribution of information in a message) and their effect on the performance of message understanding systems. The second study was to look at the the effect of discourse processing, specifically coreferencing, on the performance of message understanding systems.

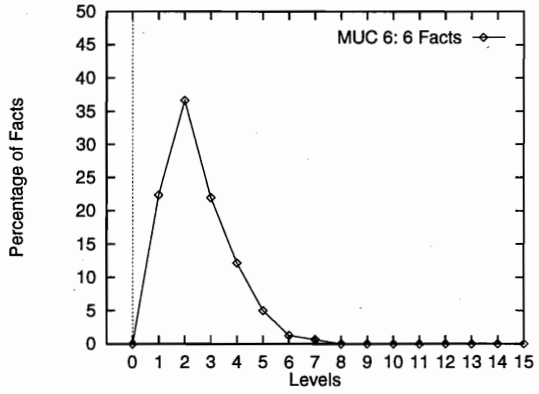


Figure 7: MUC-6: Level Distribution of the Six Facts Combined

MUC	Domain	Domain Numbers (in LUs)
MUC-4	Terrorist Attacks	1.87
MUC-5	Joint Ventures	2.67
MUC-6	Changes in Management Personnel	2.47

Figure 8: Domain Numbers of MUC-4, MUC-5, and MUC-6

9.1 Analysis of the Performance of Information Extraction Systems

We continued our analysis by examining the templates produced by the BBN, NYU, and SRI systems for the MUC-4 TST3 set of articles. We studied each template and then examined the performance of each system as it extracted the five *standard* facts for the domain. The performance of the three systems across the different levels of the five facts is shown in Figures 9, 10, and 11. The figures show the degradation in the performance of all the three systems on higher level facts. The significance of the data diminishes greatly for levels bigger than 4 because of the sparsity in the occurrence of these facts.

This type of analysis forms the basis for providing greater insight into the performances of information extraction systems. For example, a low performance on level-1 facts certainly points to problems in parsing and basic pattern training for a message understanding system. The main reason being that usually no coreferences have to be resolved when retrieving a level-1 fact. Therefore, when retrieving such a fact, a system only has to recognize patterns in the text. And inability to recognize these patterns points to problems in parsing (assuming that the system has been adapted to the domain well).

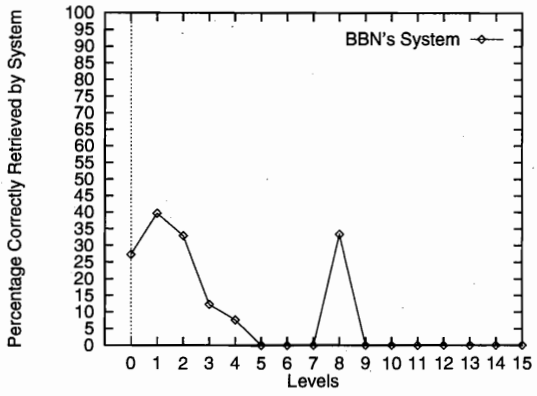


Figure 9: Performance of BBN's MUC-4 System

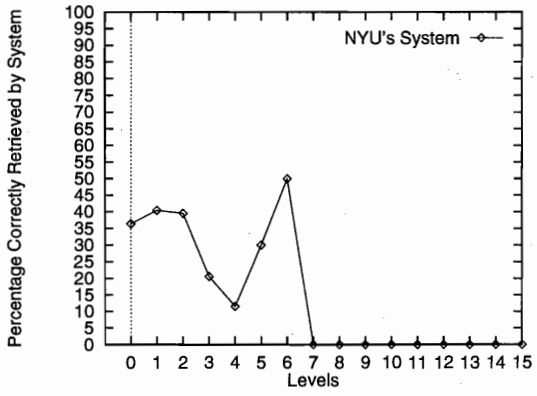


Figure 10: Performance of NYU's MUC-4 System

On the other hand, a low performance on higher (≥ 2) level facts points to problems in basic pattern training and the coreferencing module. As mentioned earlier, a level-k fact is a union of k level-1 facts. Therefore, when retrieving such a fact, a system has to identify each of the k components and then the coreferencing module has to piece these k facts together.

More details on such an analysis can be found in (Bagga, 1997).

9.2 The Role of Coreferencing

We decided, for each level, to calculate the number of coreferent nodes that comprised facts at that level. We also wanted to analyze the performances of message understanding systems based on the number of coreferences present in the facts retrieved by such a system. The analysis was using data from MUC-4 and MUC-6.

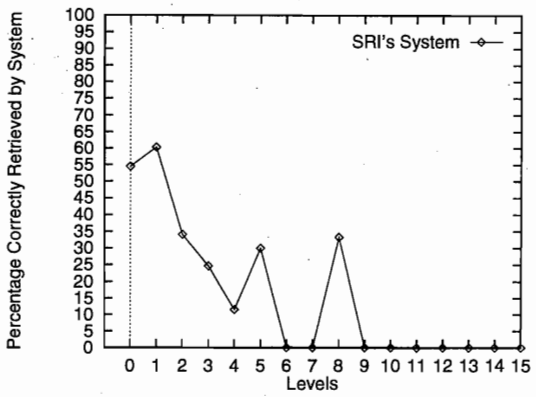


Figure 11: Performance of SRI's MUC-4 System

9.2.1 Analysis of MUC-4

For each *standard* fact at a particular level, we calculated the number of coreferent nodes that comprised the fact at that level. Figure 12 shows, for each level, the number of coreferences for all the *standard* facts at that level. Figure 13 shows the number of coreferences for all the levels combined. Because of data sparsity, the significance of the data diminishes greatly for the number of coreferences ≥ 2.

A closer look at the curves for each level in Figure 12 shows that as the level number increases, the percentage of facts having a larger number of coreferent nodes increases. For example, the curves for levels 0, 1, 2, and 3 peak when the number of coreferences equal 0, the curves for levels 4, 5, and 6 peak when the number of coreferences equal 1, and the curve for level 7 peaks when the number of coreferences equal 2. This is to be intuitively expected.

9.2.2 Analysis of the Three Systems

We analyzed the performances of the three systems on the standard facts. The performances of the three systems for all levels is shown in Figure 14.

As expected, the performances of all the three systems take a hit on facts that contain a larger number of coreferences. This confirms the results of the exploratory study done by Beth Sundheim. Moreover, the performances of the three systems on facts that had no coreferences is almost the same as their performances on level-1 facts. This is not surprising at all since most level-1 facts have no coreferences.

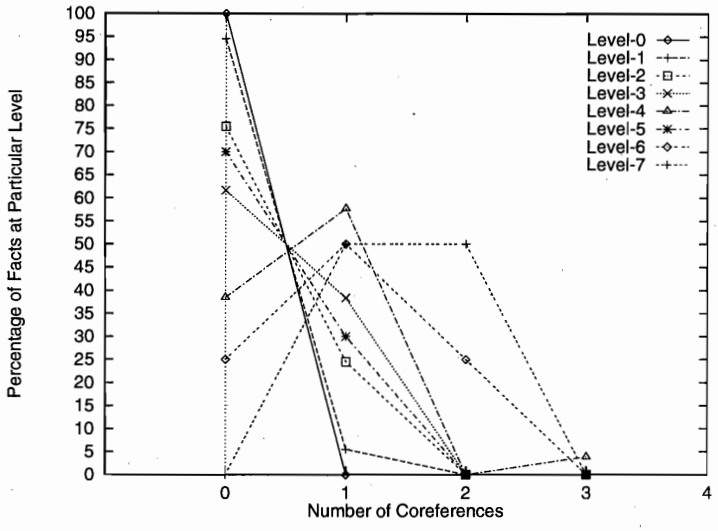


Figure 12: MUC-4: Number of Coreferences At Each Level

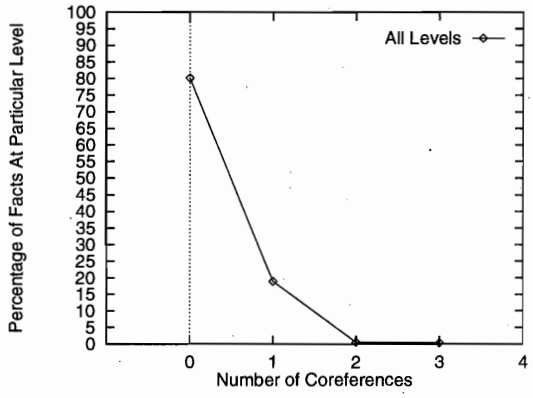


Figure 13: MUC-4: Number of Coreferences At All Levels

9.2.3 Analysis of MUC-6

As with MUC-4, for each *standard* fact at a particular level, we calculated the number of coreferent nodes that comprised the fact at that level. Figure 15 shows, for each level, the number of coreferences for all the *standard* facts at that level. Figure 16 shows the number of coreferences for all the levels combined. Because of data sparsity, the significance of the data diminishes greatly for the the number of coreferences ≥ 3.

Once again, a closer look at the curves for each level in Figure 15 shows that as the level number increases, the percentage of facts having a larger number of coreferent nodes increases (the curves for levels 1 and 2 peak when the number of coreferences equal 0, the curves for levels 3, 4, and 5 peak when the number of coreferences equal 1, and the curve

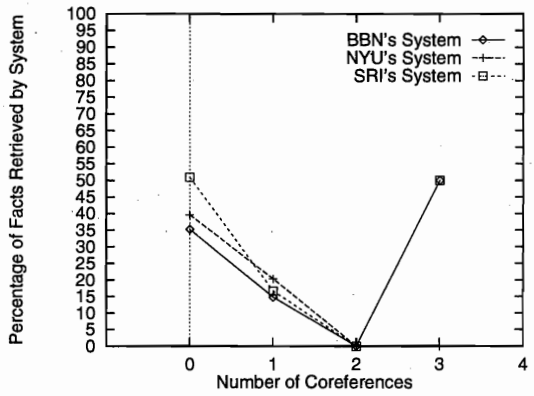


Figure 14: MUC-4: Performance of the Three System

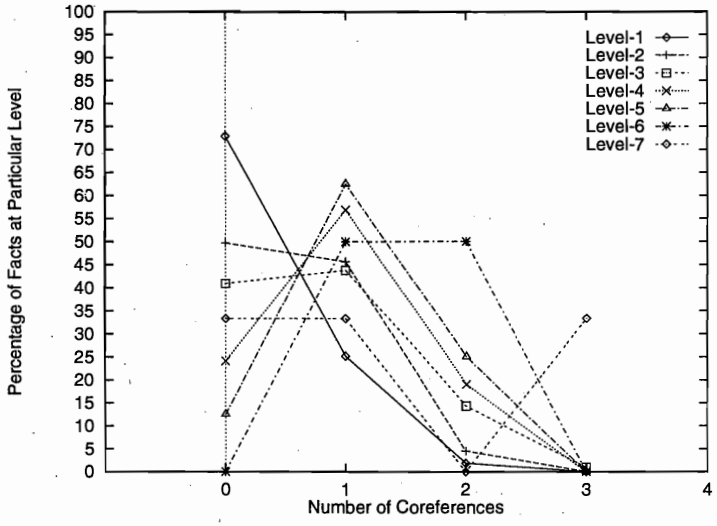


Figure 15: MUC-6: Number of Coreferences At Each Level

for level 6 peaks when the number of coreferences equal 2).

9.2.4 Analysis of The Three Systems

We analyzed the performance of the three systems on the standard facts. The performances of the three systems for all levels is shown in Figure 17. As before, the performances of the systems take a hit on facts that contain a larger number of coreferences.

Comparing Figure 14 with Figure 17 one can see that the performances of the systems on facts containing larger number of coreferences has improved considerably since MUC-4. This is a result of realization of the importance of discourse processing. It is also the result of a conscious effort on the part of the people organizing the MUCs to get the groups developing the systems to focus on discourse processing (specifically coreferencing).

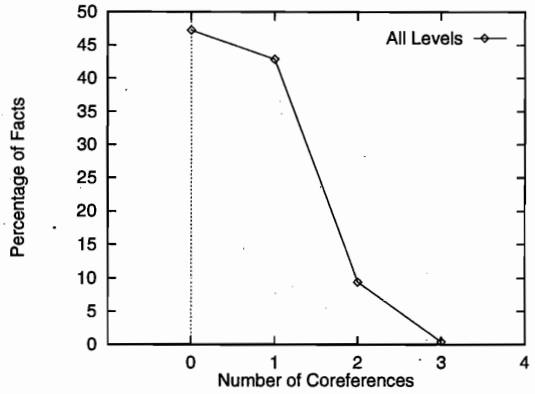


Figure 16: MUC-6: Number of Coreferences At All Levels

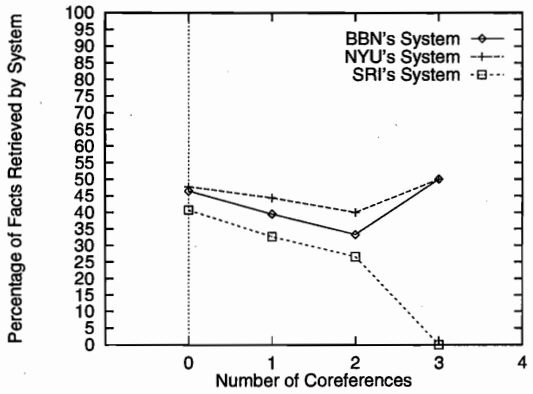


Figure 17: MUC-6: Performance of the Three System

Coreferencing was introduced as a formal (although optional) task in MUC-6. And a number of groups undertook efforts to specifically improve their coreferencing modules.

But, the surprising fact about the performances of the three systems for MUC-6 is that the hit taken because of the increase in the number of coreferences is approximately the same (Figure 17). This shows that while improvements in the coreferencing modules have helped the systems perform better, the improvements have been almost the same for the three systems. The basic difference in the performances of the three systems has stemmed mainly from their performances on level-1 facts (facts with almost no coreferences). Therefore, for information extraction systems to achieve recall and precision of 70% or higher, there has to be significant improvements in their ability to process discourse.

10 Conclusion

The level of a fact with respect to a semantic network for a piece of text provides a new method of classifying a fact based on the degree of difficulty of extracting it from that text. The analysis of the degree of difficulty of understanding a text in a domain comes as a by-product of our approach and is a big step up from some of the techniques used earlier.

11 Acknowledgments

We wish to thank Beth Sundheim for providing us with the official data from the MUCs.

References

Bagga, Amit. Analyzing the Performance of Message Understanding Systems, To Appear.

Hendrix, Gray G. Encoding Knowledge in Partitioned Networks. In *Associative Networks.* Nicholas V. Findler (ed.). New York: Academic Press, 1979, pp. 51-92.

Hirschman, Lynette. An Adjunct Test for Discourse Processing in MUC-4, *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pp. 67-77, June 1992.

Proceedings of the Third Message Understanding Conference (MUC-3), May 1991, San Mateo: Morgan Kaufmann.

Proceedings of the Sixth Message Understanding Conference (MUC-6), November 1995, San Francisco: Morgan Kaufmann.

Schubert, Lenhart K., et. al. The Structure and Organization of a Semantic Net for Comprehension and Inference. In *Associative Networks.* Nicholas V. Findler (ed.). New York: Academic Press, 1979, pp. 121-175.

Sundheim, Beth M. TIPSTER/MUC-5 Information Extraction System Evaluation, *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, pp. 27-44, August 1993.

Sundheim, Beth M. Overview of Results of the MUC-6 Evaluation, *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, pp. 13-31, November 1995.

Human Judgment as a Basis for Evaluation of Discourse-Connective-based Full-text Abstraction in Chinese

Benjamin K T'sou, Hing-Lung Lin, Tom B Y Lai
Language Information Sciences Research Centre
City University of Hong Kong
83 Tat Chee Avenue, Kowloon Tong
Kowloon, Hong Kong
rlbtsou@cpccux0.cityu.edu.hk

Abstract

In Chinese text, discourse connectives constitute a major linguistic device available for a writer to explicitly indicate the structure of a discourse. This set of discourse connectives, consisting of a few hundred entries in modern Chinese, is relatively stable and domain independent. This paper attempts to demonstrate the validity of using discourse connectives in full-text abstraction by means of an evaluation method, which compares human efforts in text abstraction with the performance of an experimental system called ACFAS. Specifically, our concern is about the relationship between the perceived importance of each individual sentence as judged by human beings and the sentences containing discourse connectives within an argumentative discourse.

1. Introduction

Through increasingly convergent interests and cross-fertilization in linguistics and computer science, research into discourse in natural language processing (NLP) has made much progress in the last decade. Discourse as understood by linguists refers to any form of language-based purposeful communication involving multiple sentences or utterances. The most important forms of discourse of interest to NLP are text and dialogue. While textual discourse normally appears as a linear sequence of sentences, it has long been recognized by linguists that these sentences tend to cluster together into units, called discourse segments, that are related in some way to form a hierarchical structure.

In NLP, discourse analysis must go beyond sentence-based syntactic and semantic analysis. Its functions are to divide a text into discourse segments and to recognize and re-construct the discourse structure of the text as intended by its author [Allen 1995]. Results of discourse analysis can be used to resolve many important NLP problems such as anaphoric reference [Hirst 1981], tense and aspect analysis [Hwang 1992], intention recognition [Grosz 1986, Litman 1990] and text generation [McKeown 1985, Lin 1991], etc.

Discourse analysis is also applicable to text abstraction, as demonstrated in Project ACFAS (Automated Chinese Full-text Abstraction System), which is an on-going, computational linguistics research project at the City University of Hong Kong. ACFAS aims to automatically produce abstracts from Chinese newspaper editorials in Hong Kong [T'sou 1992, T'sou 1996] through a new approach based on analyzing the rhetorical structure of argumentative discourse. This process, called Rhetorical Structure Analysis (RSA) [T'sou 1996], is based on the Rhetorical Structure Theory developed by Mann and Thompson for describing the discourse structure of English text [Mann 1986]. A similar approach has been made for Japanese [Ono 1994].

As a brief review of the RSA, please note that in an argumentative discourse, the progression of reasoning commonly involves *explicit* discourse connectives, that are used to express the temporal, causal or rhetorical relationships amongst constituent propositions or clauses. RSA makes use of those discourse connectives appearing in a Chinese text to (1) extract every rhetorically connected discourse segment of the text, and (2) recognize and construct the rhetorical structure of each discourse segment. Using these resultant rhetorical structures, an appropriate abstract may be generated by systematic rhetorical structure reduction to produce abstracts with differential coverage of the details of the underlying argumentation [T'sou 1996].

In modern Chinese text, discourse connectives constitute a major linguistic device available to a writer to explicitly indicate the structure of a discourse. Examples of Chinese discourse connectives include 因此("therefore"), 因爲("because"), 如果("if")...就("then"), 假如("assuming")...那末("then"), 雖然("although")...但是("but"), etc. This set of discourse

connectives, consisting of a few hundred entries, is relatively stable in modern Chinese, and is independent of the domain of discourse.

Initial corpus analysis [Ho 1993] has indicated that about 30% of sentences in Chinese editorials contain discourse connectives, which provide a key to the basic understanding of the inherent logical structure within the argumentative discourse. They also provide a potentially useful approach for scaleable and domain-independent full-text abstraction as demonstrated in [T'sou 1996]. Because the flow of argumentation is not exclusively demarcated by discourse connectives, the validity and robustness of this approach require empirical comparison with human efforts in abstraction, which can contribute to the design of a general evaluation method for automatic abstraction in Chinese. Such a comparison would entail human subjects performing abstraction on the same editorials as ACFAS and comparing their results (see also [Watanabe 1996]). Two major questions require answers from carefully designed experiments: (1) Is there relative consistency in human abstraction? (2) Is the existence of discourse connectives a relevant factor in determining the relative importance of the constituent discourse segments?

2. Design of Experiment

A set of 10 Chinese editorials was taken from two well-known newspapers published in Hong Kong and denoted as {E1, E2, …., E10}. These editorials were concerned with controversial events which occurred in Hong Kong. They included the decision to build a nuclear power plant near Hong Kong, the relationship between debt and corruption in the police force, the unemployment rate of young people, the law and the attitude of the population towards anti-discrimination, etc. These editorials are arche-typical examples of argumentative discourse.

The subjects of experiment included three groups of 25 students each from three prestigious universities in northern China. Two groups were from Chinese departments and one from a computer science department, and all were either final year undergraduates or first year graduate students. They undertook the experiment separately in time and location, and, as far as we can ascertain, these were independent experiments. The subjects were generally brought up in primarily monolingual settings and could understand the issues discussed in the

selected editorials, but, without the intimate knowledge, as well as prejudice, of the related background. It was our conscious decision to use Hong Kong newspaper editorials with Mainland Chinese subjects of above-average linguistic competence and intellectual capacity for performance comparison.

Computer print-out instead of the original texts were given to the subjects of this experiment to avoid any confusion and hints preserved in the format of the original texts. The experiments took place under controlled environment in an invigilated classroom setting.

Subjects were given the 10 selected editorials in one batch. They were asked to determine which clauses or sentences in each given editorial contained the most essential information from the author. Subjects were required to work on the editorials sequentially and in prescribed time. Each subject was asked to (1) underline in red about 10% of text which, according to his/her own judgment, contained the most important information (called *key propositions* below) in the editorial, and (2) underline in blue about 15% more of the next most important parts (called *important propositions* below) of the editorial. Subjects were specifically advised to cover as widely as possible (subject to the above constraints, of course) all aspects of the content that the author might have intended to convey.

3. Method of Analysis and Evaluation Metrics

Data analysis of the experimental results as well as performance evaluation of ACFAS were carried out as follows: (1) Target abstracts were generated per editorial per student group according to how the editorial text was marked by the human subjects. (2) Target abstracts for the same editorial were analyzed for similarity and consistency among the three groups. (3) Abstracts generated by ACFAS were compared with the corresponding abstracts generated by the human subjects according to two performance metrics, recall and precision, as defined in Section 3.2.

3.1 Generation of the target abstract

The objective of this step is to select part of a given source text to form a target abstract. The selection criterion is based on how the text is marked by the human subjects of experiment.

(i) Let WK be the weighting factor assigned to a *key proposition* and

WI be the weighting factor assigned to an *important proposition*,

where $0 < WK, WI \leq 1$.

We can compute the weighted average of the j^{th} proposition, denoted as PERC-IMP_j (for *Perceived Importance*), according to the following formula:

$$\text{PERC-IMP}_j = \frac{1}{n} \{ (\sum_{i=1}^{n} KEY_{ij}) * WK + (\sum_{i=1}^{n} IMP_{ij}) * WI \}$$

where n is the number of subjects,

$$KEY_{ij} = \begin{cases} 1 & \text{if the } j^{th} \text{ proposition is marked by the } i^{th} \text{ subject as a key proposition,} \\ 0 & \text{otherwise} \end{cases}$$

$$IMP_{ij} = \begin{cases} 1 & \text{if the } j^{th} \text{ proposition is marked by the } i^{th} \text{ subject as an important proposition,} \\ 0 & \text{otherwise} \end{cases}$$

(ii) For a given source text, we can sort all propositions of the text according to their perceived importance.

Let α ($0 < \alpha \leq 1$) be the threshold value used to separate those propositions that should be included in the *target abstract* (for PERC-$IMP_j \geq \alpha$) and those that should be excluded (for PERC-$IMP_j < \alpha$). Note that α is introduced to account for the fact that, when we talk about abstraction of a source text, there is a whole spectrum of possible abstracts with different sizes, each corresponds to a different value of α.

For a given α, we can define the *abstract ratio*, β, of the target abstract to be

$$\beta(\alpha) = \frac{\text{size of target abstract}(\alpha)}{\text{size of source text}}$$

3.2 Performance metrics for a text abstraction system

ACFAS is an experimental text abstraction system that is capable of generating multiple abstracts with differential coverage of a source text [9]. We consider only the abstract generated by the top-level output of ACFAS. We define the abstract to source ratio of the top-level output of ACFAS to be

$$\text{ACFAS-RATIO} = \frac{\text{size of top-level abstract of ACFAS}}{\text{size of source text}}$$

The following two performance measures for ACFAS are defined:

$$\text{RECALL}(\beta) = \frac{\text{\# of target propositions generated by ACFAS}}{\text{size of target abstract}}$$

$$\text{PRECISION}(\beta) = \frac{\text{\# of target propositions generated by ACFAS}}{\text{size of abstract generated by ACFAS}}$$

Note that in the above definitions, we explicitly indicate that both RECALL and PRECISION depend on the abstract ratio β of the target abstract that we choose to conduct an evaluation.

4. Similarity Analysis of Human-Generated Abstracts

In this section, results of the experiment described in Section 2 above are analyzed within the framework set out in Section 3 to examine consistency in abstracts generated by different groups of human subjects.

Text abstraction is the process of condensing salient information from a source text. It involves sophisticated and intelligent manipulation of given and assumed world knowledge as well as knowledge of natural language. It is well known that abstracts produced by different human individuals for the same source text can vary depending on the background and education level of the individuals involved. Furthermore, even for the same individual, different abstracts can be generated at different times [Luhn 1958]. While this is true with respect to the behavior of individual human beings, when they are examined as a group, our

results below show that abstracts produced by different groups of human subjects with similar educational background are in fact relatively consistent.

Fig. 1 shows the average Perceived Importance scores for the 65 propositions in one of the test editorials in respect of each group of subjects. The two weighting factors are set to be WI=0.8 and WK=1. These two values are chosen to reflect the fact that key propositions and important propositions constitute top 10% and the next 15%, respectively, of the source text according to the instruction given to the subjects of experiment.

Inspection of the three plots of Fig. 1 reveals that while there is a considerable variation in the (three) absolute scores of each of the individual propositions, the overall shapes of the three plots are obviously similar.

The similarity of the plots is statistically assessed by considering each of the propositions as an observation point. For the sake of convenience, the scores given by the 25 subjects in a group are averaged, so that there are 3 scores for each of the observation points. Pearson coefficients of correlation (pair-wise) of the (averaged) scores of the three groups calculated from data for 379 propositions in 5 common test editorials are given below.

	Group 1	Group 2	Group 3
Group 1	1		
Group 2	0.886077	1	
Group 3	0.914838	0.945098	1

As shown above, the correlation coefficients are positive and close to 1. They clearly establish strong consistency amongst the three groups of human subjects with respect to the perception of relative importance of individual propositions in the editorials. Besides confirming that human subjects do indeed generate abstracts in a consistent manner, the above analysis can also be seen as empirical evidence of the validity of the Perceived Importance score suggested in Section 3.

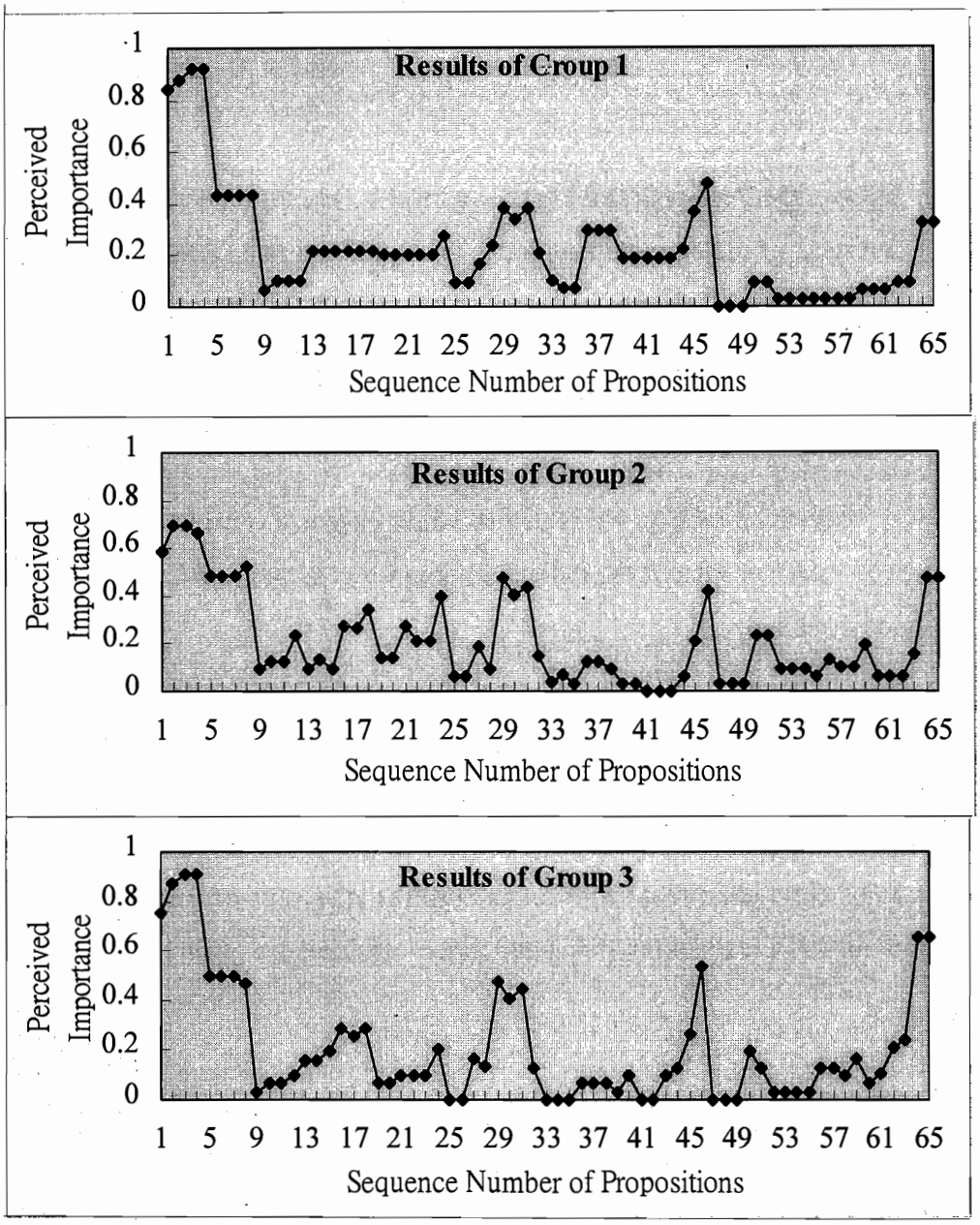


Figure 1 Perceived Importance of an Editorial for Three Groups of Subjects

5. Performance Evaluation of ACFAS: An Empirical Study

In the previous section, we have demonstrated that abstracts generated by different groups of human subjects exhibit a high degree of similarity. Therefore, it seems appropriate to evaluate the performance of a text abstraction system by comparing its output with target abstracts produced by human subjects based on the metric of Perceived Importance. In this section, we report on an empirical study of the performance of ACFAS based on the performance measures of RECALL and PRECISION defined in Section 3. This evaluation

was conducted by comparing abstracts generated by ACFAS with those target abstracts produced by the group of 25 computer science students.

5.1 Statistics on the target abstracts of 10 source texts

The average target abstract ratio's of 10 editorials, given as a function of the Perceived Importance threshold, are shown in Figure 2. The two weighting factors are set to be WI=0.8 and WK=1 as discussed above. On the average, only 12.5% of the contents of any source text has received a Perceived Importance of 0.5 or above. This indicates that, within any text, there exists a small, identifiable group of propositions which contains the most important information relevant to the text. This small group of propositions will form the basis of any abstract produced by human subjects.

On the other hand, it may be noted that about 40% of the content of any source text has received a Perceived Importance of less than 0.1. This very likely indicates a high degree of redundancy in human compositions of this genre.

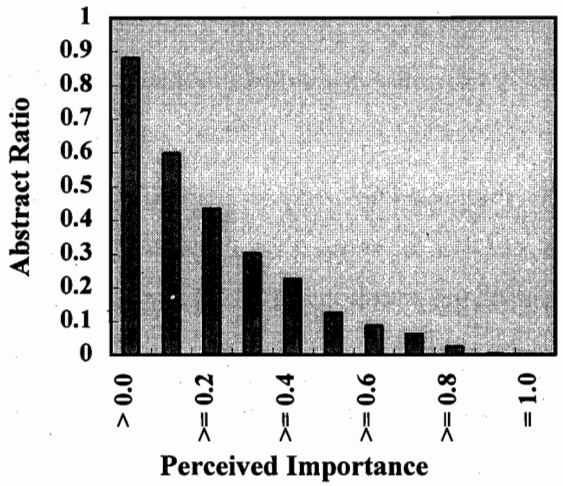


Figure 2 Abstract Ratio as Function of Perceived Importance

5.2 Statistics on the top-level abstract of ACFAS

On the average, the size of a top-level abstract generated by ACFAS is 27.4% of the source text. This is significantly higher than the target abstract ratio of 12.5% (for $\alpha \geq 0.5$) produced by human subjects. This result may be caused by the lack of explicit discourse connectives

to determine the relationships between different (yet related) discourse segments. An in depth study of more general types of discourse connectives, including explicit and implicit ones, should improve the present situation.

5.3 Performance evaluation of ACFAS

The average RECALL and PRECISION of the 10 abstracts generated by ACFAS according to how well they correspond with the target abstracts produced by human subjects are shown in Fig. 3 and 4.

As shown in Fig. 3, when the abstract ratio (i.e. the human-generated abstract size as a percentage of the source text) equals to 100%, the average RECALL is 27.4%, which is also the size of the top-level abstract generated by ACFAS. As the value of the abstract ratio reduces, the average RECALL increases modestly until it reaches a maximum value of 36.5% for the abstract ratio of 30%. This improvement of about 10% for the average RECALL is an indication of an inherent relationship between the mechanism of ACFAS and the process of human text abstraction.

Note that when the abstract ratio of 30% further reduces, the average RECALL decreases rapidly. As our abstract ratio is computed by sorting all propositions of the text according to their perceived importance, a small abstract ratio corresponds to the set of propositions that have received high average scores of perceived importance. This result indicates that ACFAS is unable to retrieve some of the most important propositions from the text. After examining the content of the source texts, we find that there is a high probability of finding important propositions in the beginning and the end of these texts (this seems to reflect a typical pattern in argumentative discourse, i.e. problem statement in the beginning and conclusion in the end of a text), but there are relatively few discourse connectives found in this area.. The present strategy of ACFAS is to ignore sentences without explicit discourse connectives between them, therefore, those target propositions located in the beginning and the end of the text will not be included in the ACFAS-generated abstract.

Fig. 4 contrasts the values of RECALL and PRECISION, both as functions of the abstract ratio. We observe that at the maximum RECALL of 36.5%, the average PRECISION

is 39.4%. In other words, about 60% of the target propositions are not extracted by ACFAS, and most of them are propositions located in the beginning and end of the source texts.

The conclusion we can draw from this result is that a system like ACFAS, which uses only the existence of explicit discourse connectives to determine the relative importance of the propositions in an argumentative discourse, performs well in the part of text that deals with the argumentative flow and presentation of evidence, but performs poorly where the problem statement is delineated and the conclusion or summarization is presented. Other factors and cues must be used to account for this deficiency.

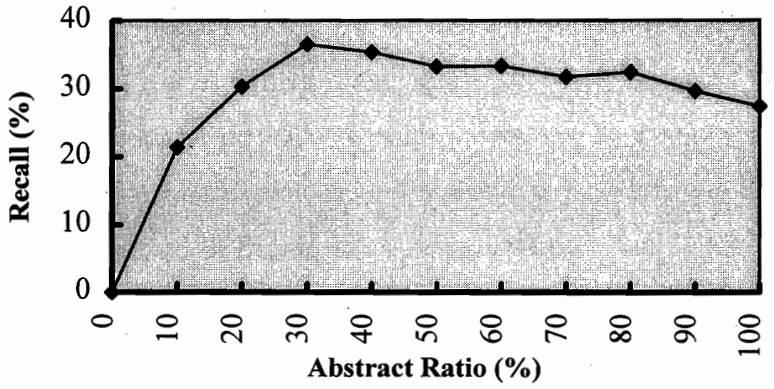


Figure 3 Recall as Function of Abstract Ratio

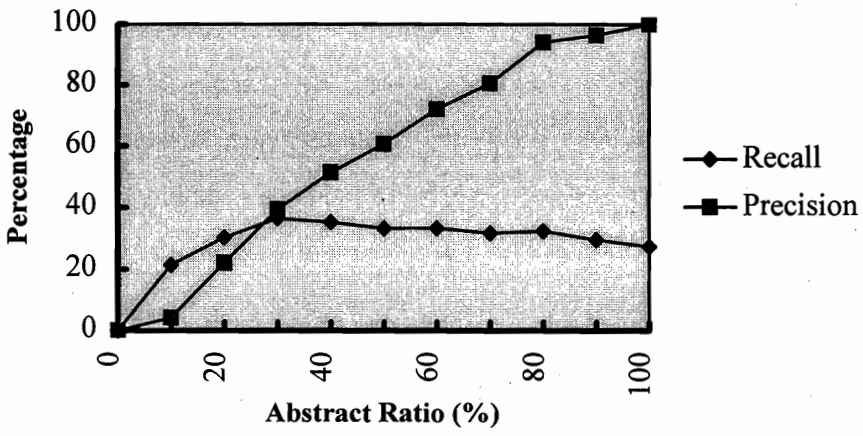


Figure 4 Recall vs. Precision as Functions of Abstract Ratio

6. Conclusions

Text abstraction entails the process of determining which sentences in a text contain the most important information that the author intends to convey to his readers. Our empirical study shows that this set of essential sentences consists of a relatively small fraction of the original text. Based on their comprehension of the text, human subjects, *behaving as a group*, are able to pinpoint this set of sentences relatively easily and consistently.

ACFAS is an automated Chinese full-text abstraction system, which extracts essential sentences from a given text following the analysis of its discourse structure. This process of ACFAS relies mainly on the presence of various discourse connectives in the text. By comparing the sentences identified as important by ACFAS with those identified by human subjects, *who presumably use additional cues*, our study shows that there is a non-random correspondence between these two sets of sentences. Since ACFAS, in its current design, does not include deep semantic processing to understand the meaning of each sentence in a text, we can conclude as follows: Which information in a text perceived by its readers as important depends not only on its semantic content, but also on how it is presented in a text, i.e. its discourse structure.

As a final remark, text abstraction represents a unique human faculty, which involves intelligent manipulation of given and assumed knowledge and natural language. Therefore, it is our belief that no single factor can guarantee its successful execution. Relevant factors or cues that had been used in the design of automated text abstraction systems include keywords, word frequency counts, discourse connectives, rhetorical relations, tense, distance from the beginning and the end of a text, just to name a few. However, there is a general negligence of systematic and quantitative evaluation of the relative contribution of each individual factor to the whole process of text abstraction. The present paper, by concentrating on the factor of explicit discourse connectives within a text, is a step toward improving this situation.

References

Allen, J., *Natural Language Understanding, 2nd Edition*, Reading, Benjamin/Cummings, Redwood City, CA, 1995.

Grosz, B.J. and C. Sidner, "Attention, Intention, and the Structure of Discourse," *Computational Linguistics* 12:3, 1986, pp.175-204.

Hirst, G., "Discourse Oriented Anaphoral Resolution in Natural Language Understanding: A Review," *Computational Linguistics* 7:2, 1981, pp. 85-98.

Ho, H.C., B.K. T'sou, Y.W. Chan, B.Y. Lai and S.C. Lun, "Using Syntactic Markers and Semantic Frame Knowledge Representation in Automated Chinese Text Abstraction," in *Proc. 1st Pacific Asia Conf. On Formal and Computational Linguistics*, Taipei, 1993, pp. 122-131.

Hwang, C.H. and L.K. Schubert, "Tense Trees as the 'Fine Structure' of Discourse," in *Proc. 30th Annual Meeting, Assoc. for Computational Linguistics*, 1992, pp. 232-240.

Lin, H.L., B.K. T'sou, H.C. Ho, T. Lai, C. Lun, C.K. Choi and C.Y. Kit, "Automatic Chinese Text Generation Based on Inference Trees," in *Proc. ROCLING Computational Linguistic Conf. IV*, Taipei, 1991, pp. 215-236.

Litman, D.J. and J. Allen, "Discourse Processing and Commonsense Plans," in Cohen et.al.(ed.), *Intentions in Communications*, 1990, pp. 365-388.

Luhn, H.P., "The Automatic Creation of Literature Abstracts," *IBM Journal of Research and Development*, 2:2, 1958, pp. 159-165.

Mann, W.C. and S.A. Thompson, "Rhetorical Structure Theory: Description and Construction of Text Structures," in Kempen(ed.) *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, 1986, pp. 279-300.

McKeown, K.R., "Discourse Strategies for Generating Natural-Language Text," *Artificial Intelligence* 27:1, 1985, pp. 1-41.

Ono, K., K. Sumita and S. Miike, "Abstract Generation based on Rhetorical Structure Extraction," *Proc. Coling'94*, 1994, pp. 344-348.

T'sou, B.K., H.L. Lin, H.C. Ho and T. Lai, "From Argumentative Discourse to Inference Trees: Using Syntactic Markers as Cues in Chinese Text Abstraction," in *Proc. 3rd International Conf. On Chinese Information Processing*, Beijing, China, 1992, pp. 76-93. Also appeared in C.R. Huang, K.J. Chen & B.K. T'sou (ed.) *Readings in Chinese Natural Language Processing*, Monograph Series No. 9, Journal of Chinese Linguistics, 1996, pp. 199-222.

T'sou, B.K., H.L. Lin, H.C. Ho, T. Lai and Terence Chan, "Automated Chinese Full-text Abstraction Based on Rhetorical Structure Analysis," *Computer Processing of Oriental Languages* 10:2, 1996, pp. 225-238.

Watanabe, H., "A Method for Abstracting Newspaper Articles by Using Surface Clues," *Proc. Coling'96*, 1996, pp. 974-979.

An Assessment on Character-based Chinese News Filtering Using Latent Semantic Indexing

Shih-Hung Wu, Pey-Ching Yang, Von-Wun Soo
Department of Computer Science
National Tsing Hua University
Hsin-Chu 30043 Taiwan R.O.C.

e-mail: *dr828307@cs.nthu.edu.tw, mr854359@cs.nthu.edu.tw, soo@cs.nthu.edu.tw*

Abstract

In this paper, we assessed the Latent Semantic Indexing (LSI) approach for Chinese information filtering. The assessment was for Chinese news filtering agents that used a character-based and hierarchical filtering scheme. The traditional vector space model was employed as information filtering model, and each document was converted into a vector of weights of terms. Instead of using words as terms in IR denominating tradition, the terms were referred to Chinese characters. LSI captured the semantic relationship between the documents and Chinese characters. We used the Singular-value Decomposition(SVD) technique to compress the terms space into a lower dimension which achieves latent association between document and terms. We showed by experiments that the recall and precision results of Chinese news filtering by character-based approach incorporating the LSI technique into the information filtering system were satisfactory.

1. Introduction

The rapid growth of Internet precipitates the need of the Network Information Retrieval System. Most of the famous systems that assist people in locating information on the Internet such as Lycos, Infoseek, Alta Vista, WebWatcher[Armstrong 95] are designed for English in-

formation retrieval. To our knowledge, only the Csmart[Chien96] and GAIS [http://gais.cs.ccu.edu.tw/] systems are designed for Chinese information retrieval. However, information filtering is conceptually slightly different from information retrieval, we have to modify the techniques of information retrieval into information filtering. In this paper, we assessed the LSI technique for a hierarchical Chinese information filtering scheme. In particular we assess the SVD approach for Chinese news filtering, which to our knowledge has never been investigated for Chinese language.

Usenet news is one of the rich information resources on Internet, filtering out useful news among thousands of available news is a crucial problem [Lang95]. Imagine a client user who needs a software agent to automatically recommend interesting news in Chinese from the Internet. Since the news is updated every day, the traditional technique for information retrieval to retrieve news with a fixed set of database would not work. Also the task that a news filtering agent faces is to select relevant news according to the user's interest or preference from a huge amount of dynamically growing news. Belkin and Croft [Belkin 92] pointed out that one major difference between information retrieval and filtering is that: The queries in information retrieval typically represent user's short-term interests, while the user profiles in information filtering tend to represent user's long-term interests. To model user's long term interest, a user profile plays an important role in information retrieval [Mayeng 90] and filtering. Profiles can be represented in many ways and at different psychological and abstraction levels. A collection of documents in a user's personal digital library may approximate the user profile. The information filtering is a document-find-document style of information retrieval. A document that is similar to the documents in the user's personal digital library is regarded as relevant.

We adopted the vector space model [Yan 94] in our design of Chinese news filtering agents. In this model, each document is represented as a vector of weights of terms. We form each user profile by merging document vectors of the same interest category. The similarity of the incoming document vectors with the profile vectors can be computed by the cosine angles between the two vectors to determine if a document is to be filtered out.

In Chinese, there is no word delimiters to indicate the word boundaries, therefore word segmentation is a difficult task to deal with. Many proper nouns or unknown words could not be found in a word dictionary with a large vocabulary [Chien 95]. The size of Chinese character vocabulary is about 13000 among which about 5,000 characters are the most commonly used characters. However, the number of Chinese words in a document collection set can be easily up to 1,000,000. To represent the personal profile in terms of words will face the difficulty of word segmentation in Chinese [Chien 96]. We will show by experiments that without word segmentation, character-based filtering incorporated LSI can be a satisfactory information filtering method.

Filtering method incorporated LSI will possibly select relevant documents whose contents have no exactly matched keywords. This is quite different from traditional technique such as the Boolean models. The probabilistic model, Bayesian belief network Model [Turtle 91] [Ribeiro 96] shared the similar feature. The Boolean models exactly matched document's terms with the combination of the search terms specified in the query. The probabilistic models estimated the degree of relevance between documents and user query by considering the appearance frequency of certain terms in the document and the user query, together with the information about term distribution in the document collection.

Since individual terms and keywords are not adequate discriminators of the semantic content of the documents and queries, the performance of the conventional retrieval models often suffers from either missing relevant documents which are not indexed by the keywords specified in the query, or retrieving irrelevant documents which are indexed by unintended sense of the keywords in the query. Therefore, there has been great interest in text retrieval research that is based on semantics matching instead of strictly keyword matching.

Latent Semantic Indexing(LSI) using Singular-value Decomposition (SVD) is an approach to overcoming this deficiency of exact keyword matching techniques. We use truncated SVD to capture the semantic structure of word usage among certain documents, and hope this relation can be applied to other documents. Using the singular values matrix from the truncated SVD, a high-dimensional vector space representing term-document matrix is mapped to a lower dimension matrix that reflects the major concept factors in the certain

documents, while ignoring the less important factors. Terms occur in similar documents will be near in the reduced vector space. Documents may satisfy a user's query when they share terms that are closer in the reduced space. Since the reduced vector spaces are more robust indicators of the semantic meaning than individual words, the performance may be better than that of the original space.

Several papers report the use of the LSI method. Conference uses the LSI method to assign submitted manuscripts to the reviewers of the *Hypertex'91* conference based on the interests of each reviewer, a set of relevant manuscripts was sent to the reviewer[Dumais 92]. The automated assignment method achieved better matching between the reviewers and their interests than the assignment by the human experts. [Syu96] presented the technique of incorporating Latent Semantic Indexing into a neural network model for text retrieval. The performance, in terms of precision and recall, was comparable to text retrieval models.

The remainder of this paper is as follows. Section 2 provides an overview of the Latent Semantic Indexing method as applied to information retrieval, and how to use truncated SVD as a LSI approach. Section 3 briefly reviews our information filtering scheme. Section 4 reports the experimental results comparing the LSI-based model and Section 5 is the discussion and conclusion.

2. Latent Semantic Indexing method applied to information retrieval

Latent Semantic Indexing(LSI) is an extension of the vector space retrieval method. We assumed that there is some unknown "Latent" association in the pattern of terms or keywords used among documents [Dumais 92], and tried to estimate this latent association. Singular-Value Decomposition (SVD) is a technique about eigenvector decomposition and factor analysis used in statistics[Cullum 85], and Latent Semantic Indexing(LSI) using SVD is one approach to modeling the latent semantic relationships between the documents and the index terms. This approach performs singular-value decomposition on a term-by-document matrix, generating a reduced space with lower dimension. The similarity between two documents is calculated according to the index terms used in each of the documents occur in other documents. Using the LSI representation, documents satisfy a user query when they share terms of

similar semantic meaning in the reduced vector space. The dimension of the resulting vector space is much smaller than the number of exact index terms used in a document collection(e.g. from several thousands to 100 or 300[Dumais 94]), a filtering model using LSI can benefit from requiring less time and memory.

2.1 Singular-Value Decomposition(SVD) and truncated SVD

SVD is a reliable tool for matrix factorization. For any matrix A, A^TA has nonnegative eigenvalues. The nonnegative square roots of the eigenvalues of A^TA are called the singular values of A, and the number of the non-zero singular values are equal to the rank of A, *rank(A)*. Assume that A is an *m by n* matrix and *rank(A)* = r, the singular -value decomposition of A is defined as

$$A = U W V^T,$$

where the size of U is *m by m*, the size of V is *n by n* and the size of W is *m by n*. Both the U and V^T, are orthogonal matrices, i.e., $UU^T = I_m$, and $VV^T = I_n$; W is a diagonal matrix consists of the singular values of A: $\sigma_1, \sigma_2, ..., \sigma_r$. And the σ_j's are the singular values of A, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$, and $\sigma_j = 0$ for $j \geq r+1$.

To apply SVD as a LSI tool, term-by-document matrix A must be constructed. Using SVD to generating optimal approximation of the document representation specified by the matrix A. Since the singular values in matrix W are ordered from largest to smallest, the first k largest may be kept and the remaining smaller ones are set to zero. As a result, the representations of the matrices U, V, and W can be reduced by reform a new diagonal matrix W_k by removing column and rows which are zeros from W; reform a matrix U_k by removing the *(k+1)*st to the mth columns from U; and reform a matrix V_k by removing the *(k+1)*st to the nth rows from V. The product of the resulting matrices is a matrix A_k which is an approximation of the matrix A, and *rank*(A_k)=k.

$$A_k = U_k W_k V_k^T,$$

The LSI method using SVD can be viewed as a technique for deriving a set of non-correlated indexing of factors(i.e. the singular values), which represent different concepts in

the usage of words in the documents collection. The documents and queries are then represented by vectors of factor values, instead of the individual index terms. Using the *k*-largest factors may captures the most important latent semantic relation between documents and index terms, and avoids unintended sense in word usage.

2.2 Document and query representations

Since the term-by-document matrix A has been reduce to a lower dimension matrix, the vector which represent the query and all the new-add document must be mapped to the same lower dimension. Using the singular-value decomposition, a term-by-document matrix A is mapped into a reduced *k by n* matrix represented by $W_k V_k^T$, which relates k factors to n documents. A query q, originally of dimension size m, can be mapped into a size k vector q'

$$q' = (q^T U_k W_k^{-1})^T,$$

The similarity between two documents then is computed using this shorter vector representation.

3. The character-based Chinese news filtering Scheme

3.1 The character-based vector representation of documents and personal profiles

A Chinese character is the basic processing unit and is used equivalently as the concept of a "term" in IR denominating tradition, we use terms to refer to Chinese characters in the context of the paper. In our approach, no stemming and stop word lists or a thesaurus is used. We represent the weight of a term in a given document by adopting Salton's well-tested *TFIDF* formula in IR, the term frequency (tf) multiplied by the inverse document frequency (idf) [Salton 89] [Salton 91]. Namely, the weight of a term t in a given document d, namely w(t,d), is represented as

$$w(t, d) = tf_{t,d} * log(N/df_t)$$

where documents number N is the total number in a collection of documents, term frequency $tf_{t,d}$ is the number of appearance of term t in document d, and the document frequency df_t is the number of documents which content term t in the collection.

A document D can be represented as a vector V with elements $v_1, v_2, \ldots, v_n$, where n is equal to the size of character vocabulary, and vi is the weight of term i in the document. All vector are normalized, for convenience and by convention. We can calculate the similarity between two documents Di and Dj by the cosine of the angle between their vector representations:

$$\text{Similarity}(Di, Dj) = \frac{Vi * Vj}{\|Vi\| \|Vj\|}$$

where V_i is the vector representation of Di, * represents the inner product between two vectors; $\|Vi\|$ represents the norm of a vector Vi. Based on the formula, two documents with same character set will have the highest similarity between them because the inner product of the two document vectors would be one, while two documents without any character in common will have the lowest similarity zero.

We merge the document vectors in the same interest group (either grouped by the user or by a classification/clustering agent) into a higher level profile vector by their vector sum. The profile vector is also normalized.

3.2 The tasks of a news filtering agent

A Chinese character-based news filtering agent will carry with it a set of weights in terms of inverse document frequencies as discussed in section 2.2.1 for a vocabulary of terms (characters), a profile vector that represents a certain interest category and a similarity threshold associated with the profile vector. For each news document in the news server, the filtering agent will convert it first to a document vector and then the similarity between the document vector and the profile vector is computed according to the method discussed in section 2.1. If the similarity of the document with the profile is lower than the threshold, it is filtered out.

3.3 The hierarchical information filtering scheme

The hierarchical information filtering scheme reduces agent's total task. By composition of profile vectors, we reduce the number of vectors that each agent must compare with document

vectors on the web. All the lower-level profile vectors are combined to form higher-level profile vectors. We may assume that the final highest level profile vector can represent an overall interest of the user. The intelligent news filtering agent can then carry this profile vector in search for relevant documents on the web.

4. Experimentation

4.1 Data collection and document vectorization

We gathered three sets of articles from on-line China Times [http://www.chinatimes.com.tw/] for 2 consecutive weeks from Mar.2nd 1997 to Mar.15th 1997. There were 671 articles in the first week, 669 in the second week. These articles were written by professional reporters and we collected all the articles from all the nine categories that China Times provided. The categories are: *Entertainment, Sports, Economy, Focus, International, Mainland, Social, Taiwan and Editorial.*

Table 1. The number of documents in the document collection sets.

Category \ Set	1st week	nd week
Economy	86	95
Editorial	14	14
Entertainment	80	82
Focus	80	79
International	70	63
Mainland	63	58
Social	67	73
Sports	90	87
Taiwan	114	111
Total	671	669

Table 1. shows the number of documents in each of the categories. The length of each article is about 500-2000 Chinese characters. In order to test the usage of words in the document collection sets is stable or not, we use the 671 articles in the first week as the training set to compute the document frequency df_t for each term t and the 669 articles in the second week as the testing set for the filtering experiment.

The articles were first transformed into normalized document vectors as discussed in section 2.1, all English characters and Arabic numerals were ignored. The similarity between two documents is then equal to the inner product of two vectors. To mimic a user's interests, we choose news articles from three categories (*Entertainment, Sports, Economy*) on the same day (Mar. 2^{nd}) to form the initial user's profile. The user profile is treated as a set of documents and are transformed into normalized document vectors. In composition of the user profile vector we treated the importance of all articles equally.

4.2 Experiments on information filtering with SVD

To evaluate the effectiveness of news filtering based on character-based method for Chinese news document, the tf-idf weighting and vector space model were adopted in our experiment on the nine news categories, and we tried different k using truncated SVD. As discussed above, several arbitrary articles of each category in the training set were selected and merged into one *query* document. The query document was then transformed into a normalized vector named as a query vector or a profile vector. By comparing the query (profile) and document vectors in the test set, we retrieved the most similar documents in the test set and measured the precision against different recall values as plotted in **Figure 1-3**.

From **Figure 1-3**, we observed that different k number in SVD had different performance. The performance was worse either when the k number was small, e.g., 2, or when the k number was large, e.g., 100. The experiments show the suitable value for k is 10 for our document collection sets. [Dumais 94] suggested that the probable k number is from 100 to 300 in English document. Our experiments disagree with this. The great reduction of vector dimension will save a lot of memory space and time consuming on further utilization of document vectors. Before this, we perform more experiments to justify our observation.

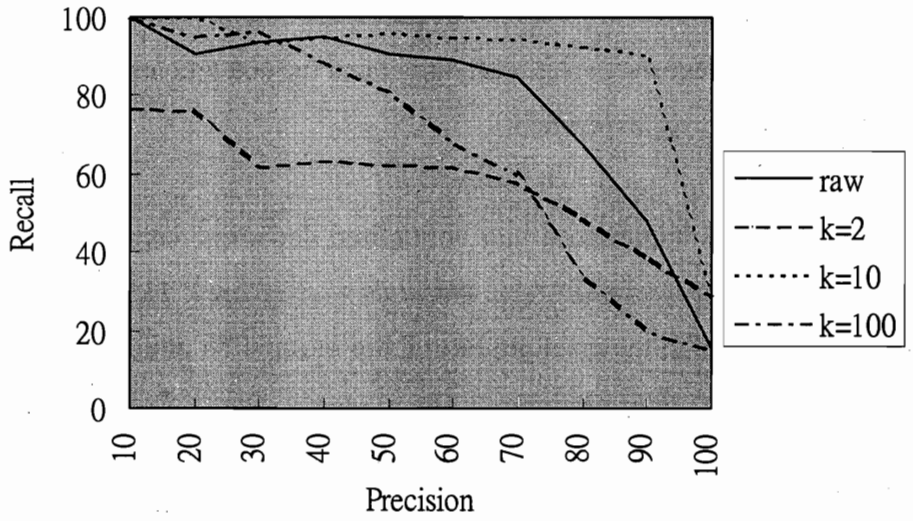


Figure 1. Recall-precision curve. Four different processing methods (raw vector form and SVD with k=2, 10, 100) on economy category.

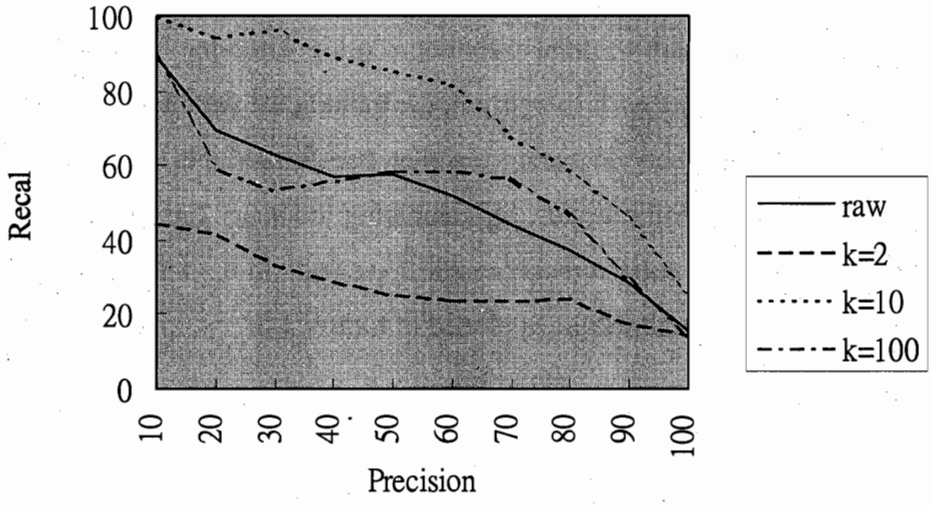


Figure 2. Recall-precision curve. Four different processing methods (raw vector form and SVD with k=2, 10, 100) on entertainment category.

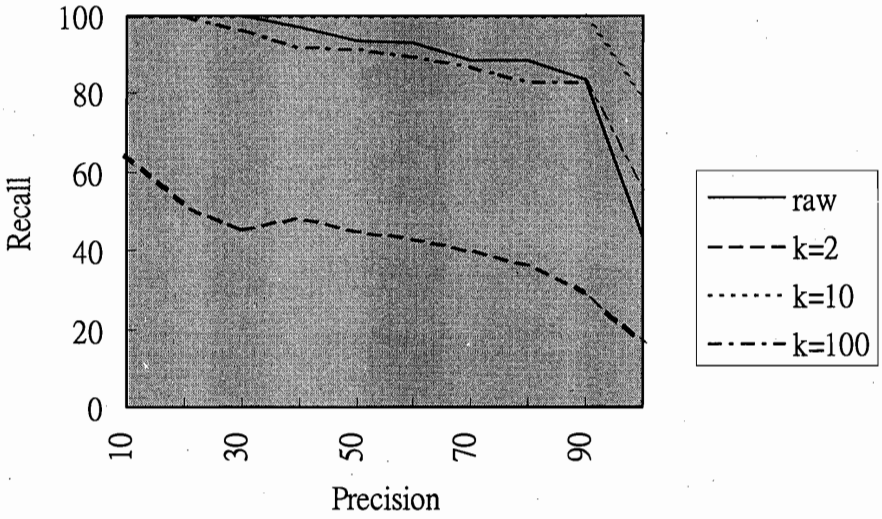


Figure 3. Recall-precision curve. Four different processing methods (raw vector form and SVD with k=2, 10, 100) on sports category.

4.3 Information filtering experiments based on different k values

To justify our observation, we tried more different k value, and calculate 11-point average precision against different k values as plotted in **Figure 4.** As in experiment 1, several arbitrary articles of each of the three categories (Sports, Economy, Entertain) in the training set were selected and merged into one *query* document. The query document was then transformed into a normalized vector named as a query vector or a profile vector. By comparing the query (profile) and document vectors in the test set, we retrieved the most similar documents in the test set, and measured performance by the 11-point average precision (average over different recall values from 0% to 100%, 10% each step).

From Figure 4, we observed that the performance reaches its maximum when k is about10 for each of the three testing profiles. The experimental result is consistent with the result in experiment 1, but quite differently from what [Dumais 94] suggested. We conjecture that the probable *k* number is different for Chinese and English and for different document

collection sets. To prove the conjecture, more experiments are needed.

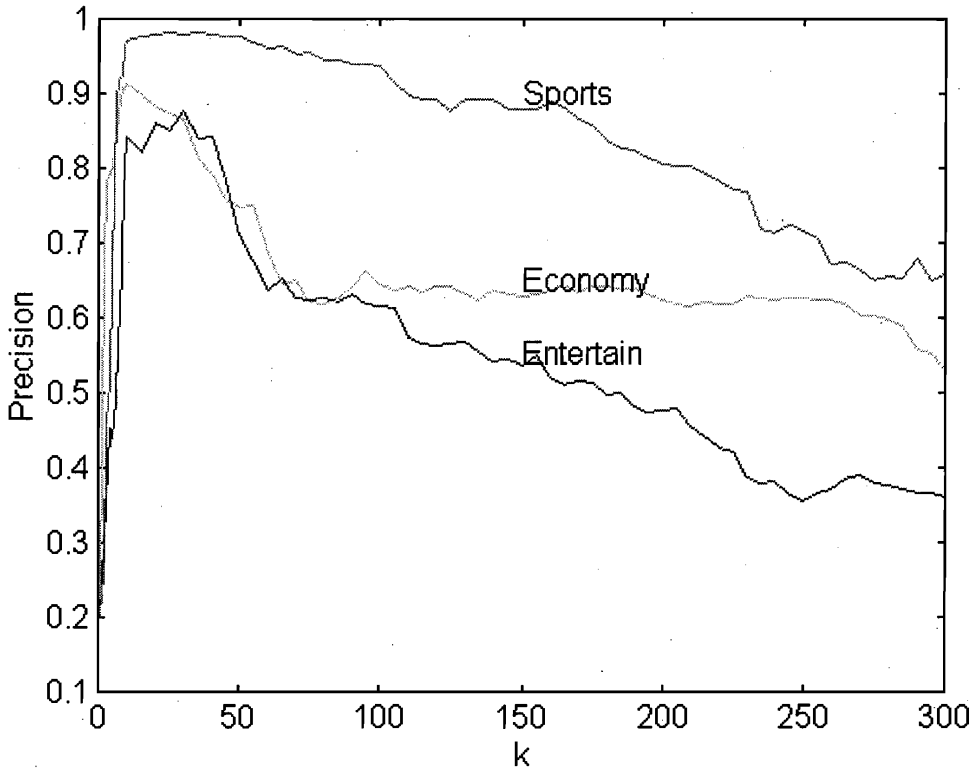


Figure 4. Performance(11-point average precision) varies against different *k* values.

5. Discussions and Conclusion

In the experimental results, we found that the recall-and-precision are surprisingly satisfactory in the character-based document-find-document style of information filtering for Chinese news filtering. The SVD technique can be used to reduce the need of storage space of the term-by-document matrix and the processing time on further utilization. The difference on performance for different choice of k when using truncated SVD value is quit interesting.

Without the word segmentation, neither stemming and stop word lists nor a thesaurus is used, the well performance of Chinese character-based information filtering is an interesting

finding. This finding experience has been also shared by Dr. Lee-Feng Chien in personal communication. This finding suggested that the semantic meaning of a Chinese news article can be implied by the character set. Articles with similar character sets tend to have similar meaning. Even though in Chinese different orders of the same set of characters may have different meaning, and the same word may have ambiguities in part of speech, character-based filtering seem to provide more information.

Character-based information filtering scheme makes a lot of sense in the sense that no dictionary of a large size of words is available and the word segmentation task in Chinese is difficult. Only about the weights and counts of most commonly used characters in the documents collection set are needed to design an intelligent news filtering agents. A truncated SVD approach with yield better performance and save more computation time for the filtering agents. The effect of SVD method is: reduce the size of the term-by-document matrix, and sort the significance of dimensions for the matrix. This should be the reason why a choice of a suitable k will give a better performance. The first k dimensions are necessary and sufficient for discriminating the categories. If we view stop words as noise, the larger the k value, the more the noise will be considered. On the other hand, the smal k may be insufficient for the discrimination among the categories.

To represent user profile and perform news filtering hierarchically not only has the merit of saving computation cost but also has the potential to perform the information filtering task in distributed and parallel manner. The efficiency will be promoted even further if each profile vector runs independently on a distributed system. This could be achieved because of the independence property among profile and document vectors, i.e., they don't interfere each other while executing similarity calculations.

In the future, the relevance feedback from the user can be used to improve performance by adjusting several system parameters. It can be used to adjust the thresholds at each stage or to adjust the weights of combining lower level profile vectors into higher level ones. We are looking into such machine learning techniques as neural networks [Pannu 95] [Syu 96] along this direction.

Acknowledgment This work is financially supported by Institute for Information Industry and National Science Council of Taiwan, Republic of China under the grant No. NSC86-2213-E-007-53.

References

Armstrong, R. and D. Freitag, T. Joachims, and T. Mitchell, "WebWatcher: A learning apprentice for the world wide web," 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, Stanford, March 1995.

Belkin, N.J. and Croft, W.B., "Information filtering and information retrieval: two sides of the same coin?," Comm. ACM 35, 12(Dec.), pp. 29-38.

Chien, L.F., "Fast and quasi-natural language search for gigabytes of Chinese texts," ACM SIGIR 95, 1995.

Chien, L.F., "An intelligent Chinese information retrieval system for the Internet," Proceedings of the ROCLING IX, 1996.

Cullum, J.K. and R.A. Willoughby, "Lanczos Algorithms for Large Symmetric Eigen value Computations - Vol. 1, Theory(Ch 5: Real Rectangular Matrices)," Brikhauser, Boston, 1985.

Dumais, S.T. and J. Nielsen, "Automating the Assignment of Submitted Manuscripts to Reviewers," Proc. Of the 15th International Conference on Research and Development in Information Retrieval, pp. 233-244, 1992.

Dumais, S.T., "Latent Semantic Indexing and TREC-2," The Second Text Retrieval Conference(TREC-2), NIST Special Publication 500-215, pp. 105-115, 1994.

Lang, K., "Newsweeder: learning to filter Netnews," Proceedings of the Twelfth International Conference on Machine Learning, 1995.

Mayeng, S. H. and R. R. Korfhage, "Integration of user profiles: models and experiments in information retrieval. Information Processing and Management," Vol. 26, No. 6, 1990.

Ribeiro, B.A.N. and R. Muntz, "A Belief Network Model for IR," In Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval, pp.253-269, 1996.

Salton, G., "Automatic Text Processing," Addison Wesley, Reading, Massachusetts, 1989.

Salton, G., "Developments in automatic text retrieval," Science 253, 1991.

Pannu, A. S. and K. Sycara, "A learning personal agent for text filtering and notification," Proceedings of the International Conference of Knowledge Based Systems (KBCS 96), Dec. 1996.

Syu, I., S. D. Lang, and N. Deo, "Incorporating latent semantic indexing into a neural network model for information retrieval," Proceedings of the Fifth International Conference on Information and Knowledge Management, Nov. 1996.

Turtle, H. and W. B. Croft, "Evaluation of an Inference Network based Retrieval Model," ACM Transactions on Information Systems, Vol. 9, No. 3, July 1991.

Yan, T. W. and H. Garcia-Molina, "Index structures for information filtering under the vector space model," Technical Report STAN CS-TR-93-1494, Nov. 1993.

INTEGRATING LONG-DISTANCE LANGUAGE MODELING TO PHONEME-TO-TEXT CONVERSION

Tai-Hsuan Ho[1], Kae-Cherng Yang[2], Juei-Sung Lin[1], Lin-Shan Lee[123]
[1]Department of Computer Science and Information Engineering, National Taiwan University,
[2]Department of Electrical Engineering, National Taiwan University,
[3]Institute of Information Science, Academia Sinica,
Taipei, Taiwan, R.O.C.
email : tai@speech.ee.ntu.edu.tw, Tel. : 886-2-369-2535

Abstract

This paper presents a phoneme-to-text conversion system for Chinese language using long-distance language modeling. First of all, we employ extended bigrams (Huang 1993) of window size *d* to capture the long-distance dependent relations in Chinese language, in which *d* bigram tables are estimated independently from the training data for distance 1 to *d*. Each bigram table is associated with a mixture weight, which can be optimized based on the held-out data using deleted interpolation algorithm (Ney 1994). The system then performs the tree-trellis search (Soong 1991) to generate N-best sentence hypotheses, and integrates these extended bigram probabilities at sentence level. In our experiments, we generate 200 best sentence hypotheses and the integration of long-distance bigram reduces the error rate by about 11% as compared with word bigram language model only. Secondly, to reduce the number of parameters, we merge the extended bigram tables from distance 2 to *d* to form a single long-distance bigram table, disregarding the influence caused by different distances. Since the model complexity is significantly reduced, we derive a very efficient stack decoding algorithm for the integration of this augmented long-distance information. Experiments show that the error rate remains the same as that of *d* extended bigrams using N-best search algorithm, while the search efficiency is significantly improved.

1. Introduction

It has been studied intensively for many years to find good approaches to input Chinese characters, for which standard keyboard is not well suited. Technologies including Speech Recognition, Hand-Writing Recognition, Optical Character Recognition (OCR) have been adopted for this task, aiming to provide users with very efficient and natural ways to input Chinese characters. However, these systems are not prevailing mainly because special hardware

is required or the technologies are not mature enough to be accepted by the users. Keyboard input methods are still the most widely used ones. There are tens of different methods for the input of Chinese using keyboard. In this paper, we confine ourselves to deriving good models and search algorithms for the phoneme-to-text conversion task, converting the input tonal syllable sequence into Chinese characters. These syllables are composed of sequences of phonemes from the standard phoneme set (ㄅ, ㄆ, ㄇ, ㄈ, etc.) and 5 tones. Each phoneme and tone is associated with a key stroke. In Chinese language, a large portion of syllables have tens of homonym characters, and some of them even have more than a hundred. The goal of this system is therefore to automatically pick the right one from the homonym set as accurate as possible.

There are a lot of related researches being explored in this topic. Among them, 漢音 (Kuo 1995) and GOING (Hsu 1994) are probably the most noticeable ones. 漢音 system uses a rule-based and statistic-based hybrid method to this problem. Morphological rules, word length heuristics, and syntactic and semantic connection tables are applied to enhance the accuracy. In GOING system, Semantic Pattern Matching based approach is adopted, which requires linguistic experts to derive a lot of templates at different levels for Chinese language. Since some templates can model Chinese language at phrase or sentence level, GOING system has been one of the systems that successfully handle the long-distance dependent relations in Chinese language. Instead of deriving costly templates and rules, here we propose a completely statistical approach to the integration of long-distance language modeling. We first apply the current language processing technology of Golden Mandarin (III) (Wang 1997) speech recognizer to solve the phonetic Chinese input problem as our baseline system, and then integrate long-distance Markov language model to enhance the accuracy.

Markov source language model has been widely adopted and proven very effective in many language processing tasks, such as speech recognition and machine translation. Given a word sequence $W_{1,n} = w_1, w_2, ..., w_n$, the language model probability for this sentence is

$$\Pr(W_{1,n}) = \prod_{i=1}^{n} \Pr(w_i | W_{1,i-1}) \qquad (1)$$

To robustly estimate the probabilities and reduce the number of parameters, n-gram language model assume the probability for the current word w_i depends only on its previous n-1 words $W_{i-n+1,i-1}$. Equation (1) thus becomes

$$\Pr(W_{1,n}) \cong \prod_{i=1}^{n} \Pr(W_i | W_{i-n+1,i-1}) \qquad (2)$$

In Golden Mandarin (III), we use n equals to 2, namely bigram, which turns out to be a good tradeoff between model complexity and performance. However, bigram fails to model the dependency for words with distance longer than 2, and this kind of dependent relations can be found quite frequently in many cases. Some examples are listed below. In these example , each sentence has been segmented properly into the corresponding word sequence, and words underlined are assumed mutually dependent.

Example 1: 洗 了 一 個 舒服 的 澡 (take a comfortable bath)

Example 2: 依 這種 法律 規定 (according to this regulation)

Example 3 : 一 隻 可愛 的 小 花 貓 (a lovely little spotted cat)

Given the phoneme sequences of these examples, n-gram with small n fails to convert them to their corresponding Chinese characters correctly. In the first example, Golden Mandarin (III) will convert the last word to another homonym "早 (morning)", since it has higher probability than the correct one "澡 (bath)". It will never be correctly converted unless the system takes into consideration the verb at the begin of the sentence "洗 (take)". Similarly in example 2 and 3, "依 (according to)" will be converted to "一 (one)" and "隻 (indefinite article for animal)" replaced by "支 (indefinite article for equipment)" if only bigram is applied.

To remedy these errors, our approach first use extended bigrams of window size d to capture the long-distance dependency. Unlike bigram language model that can be integrated efficiently by dynamic programming algorithm, long-distance model can only be integrated at sentence level by using algorithms such as N-best search. In this task, large N is necessary to avoid missing the global optimum. We use N=200 in our experiments, and the integration of extended bigrams reduces the error rate by about 11% as compared with word bigram language model only. Secondly, we merge these extended bigram tables from distance 2 to d to reduce the number of parameters, and derive a very efficient stack decoder to integrate this augmented long-distance model. Experiments show that the performance is significantly improved without degrading the accuracy. On a Pentium-Pro 200 MHz machine, the processing speed can be as high as 60 characters per second after fully optimizing the implementation.

The rest of this paper is organized as follows. Next section describes our baseline system : language processing module of Golden Mandarin (III) using word bigram. Section 3 describes

the training and estimation procedure of the extended bigrams adopted in our system, and also briefly describes the N-best search algorithm for the integration of these bigram tables. In Section 4, we smooth the long-distance information by merging the extended bigram tables from distance 2 to *d*, and derive a stack decoder for the integration of the merged long-distance model. In section 5 we show our experimental results for the algorithms and models described in previous sections, and conclude our remarks in section 6.

2. Baseline System : Language Processor of Golden Mandarin (III)

Our baseline system is the language processor of Golden Mandarin (III) speech recognizer. The input to this module is a syllable lattice, with which many confusing syllable candidates are included for each speech segment. For the phoneme-to-text conversion task, only one syllable is present at each segment. The architecture for the phoneme-to-text conversion baseline system is shown in Figure 1. Given the input phoneme sequence, a word lattice is first constructed by exhaustively looking up the lexicon for all possible word hypotheses, in which all possible transcriptions for this input phoneme sequence are encompassed. Figure 2 gives an example of the partial word lattice for the input phoneme sequence

"ㄒㄧㄢˇ ㄌㄜ˙ ㄧˊ ㄍㄜ˙ ㄕㄨˇ ㄈㄨˊ ㄉㄜ˙ ㄗㄠˇ"

A word lattice is a Direct Acyclic Graph (DAG) in which each node represents a word, and each arc represents a possible word transition. Each arc is associated with a bigram transition probability which can be estimated automatically from the training data. Word lattice is a very compact representation for all possible sentence hypotheses. Each path from the sentence start to the sentence end is a possible transcription. In this example, the best transcription for this syllable sequence is "洗 了 一 個 舒服 的 澡" (take a comfortable bath), corresponding to the path connected by the thick lines in Figure 2.

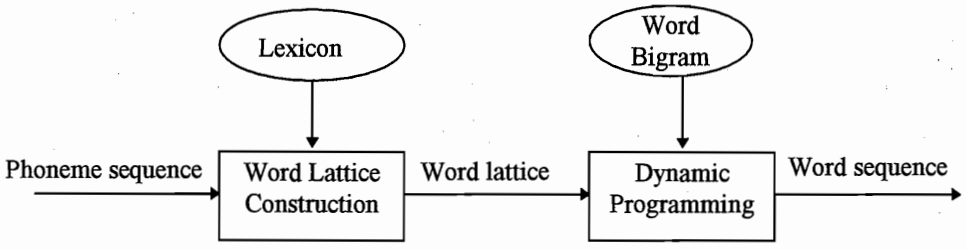


Figure 1. The architecture of the baseline system

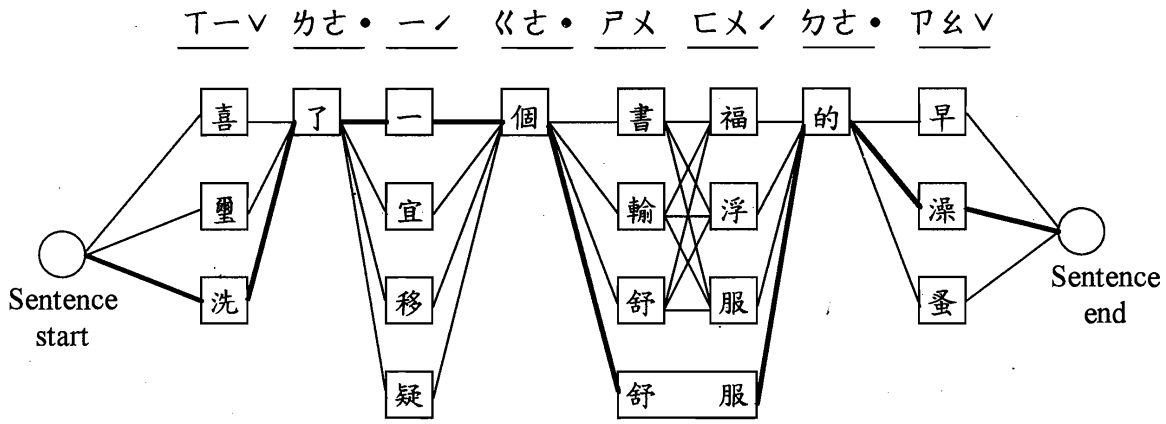


Figure 2. An example of a partial word lattice and corresponding the best path

The language model of the baseline system is word bigram. Given a path in the lattice with word sequence $W_{1,n} = w_1, w_2, ..., w_n$, the word bigram probability for this word seqauence is

$$\Pr(W_{1,n}) = \Pr(w_1) \times \Pr(w_2 | w_1) \times ... \Pr(w_n | w_{n-1}) \quad (3)$$

The bigram probability can be estimated using Maximal Likelihood Approach to maximize the total probability of the training data. However, for testing data, it may result in zero probability for word pairs not present in the training data, hence unreasonable zero probability for the whole testing data. This happens when the available training data is sparse, and can be solved by back-off to unigrams for unseen bigrams. Our back-off scheme is adopted from BBN's approach (Placeway 1993). The bigram probability for word pair $w_j w_i$ is

$$\Pr(w_i | w_j) = \tilde{\Pr}(w_i | w_j) + \bar{\Pr}(w_i | w_j) \quad (4)$$

and

$$\tilde{\Pr}(w_i | w_j) = \frac{Count(w_j w_i)}{Count(w_j) + Branch(w_j)} \qquad \text{for } Count(w_j w_i) > 0 \quad (5)$$

$$\bar{\Pr}(w_i | w_j) = \frac{Branch(w_j)}{Count(w_j) + Branch(w_j)} \times \Pr(w_i) \qquad \text{otherwise} \quad (6)$$

where *Branch(w)* is the branch factor, number of distinct succeeding words, of *w* found in the

training data. In equation (5), denominator is enlarged by its branch factor, resulting in a smaller probability for the trained word pairs. This lost probability is then distributed to all possible word pairs as their back-off bigram probabilities, and the distribution is proportional to their unigram probabilities $\Pr(w_i)$ of the current words, as described in equation (6). The back-off coefficient at equation (6) is roughly proportional to the branch factor $Branch(w_j)$ of the previous word w_j. There is a physical meaning for this back-off coefficient. If the branch factor of the previous word w_j is large, then the probability that word w_j can transit to w_i should be large, too, even though this transition was not found in the training data. If this branch factor is small, then the untrained transition probability should be small as well, since it is quite determined to transit to some specific words only.

After constructing the word lattice, the search engine will extract the best path maximize equation (3). The search engine of the baseline system contains a forward dynamic programming pass and a backtrace pass. In the forward pass, for each node w the system will compute the best partial path score from sentence start to w, α_w, using the following equation.

$$\alpha_w = \max_u \{\alpha_u + \log \Pr(w|u)\} \qquad (7)$$

where u are all words in the lattice immediately preceding w. For each word w, the best preceding word $\tilde{u}$ is recorded. In this way, the best path can be obtained by tracing $\tilde{u}$ from the sentence end all the way to sentence start in the backward pass.

3. Extended Bigrams and N-best Search Algorithm

In a traditional stochastic language model, the current word is predicted based on the preceding word (bigram) or the preceding n-1 words (n-gram). This is because most of the relevant syntactic information can reasonably be expected to lie in the immediate past. But some information, syntactic as well as semantic, may still exist in the more distant past, though use of n-gram with large n will increase the number of free parameters exponentially. To reduce the number of free parameters and maintain the modeling capacity, we use a set of extended bigrams (Huang 1993) for different distances to approximate the Markov source language model probability, as shown in the following equation :

$$\Pr(w_i | W_{1,i-1}) = \sum_{k=1}^{d} \lambda_k \Pr_k(w_i | w_{i-k}) \quad (8)$$

where $\Pr_k(w_i | w_{i-k})$ is the probability that predicts word w_i based on the word w_{i-k}. In our system, we use d=5, assuming the long-distance dependency can be ignored for distance greater than 5. In equation (8), each distance is associated with a mixture weight λ_k, which can be optimized based on the held-out data using deleted interpolation algorithm (Ney 1994) as described in the following equation.

$$\lambda_k^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} \frac{\lambda_k^{(t)} \Pr_k(w_i | w_{i-k})}{\sum_{k=1}^{d} \lambda_k^{(t)} \Pr_k(w_i | w_{i-k})} \quad (9)$$

where N is the size of the held-out data. The above equation re-estimate the next set of mixture weights $\lambda_k^{(t+1)}$ based on the current one $\lambda_k^{(t)}$. The iteration continues until the change of all the $\lambda_k^{(t)}$ can be neglected. Also, to fully utilize the available training data, we subdivide them into 5 parts, and re-estimate λ_k in a leave-one-out manner (Duda 1973).

The search algorithm follows the tree-trellis N-best search paradigm (Soong 1991) which contains 2 stages. In the first stage, the system efficiently generates N-best sentence hypotheses using word bigram only, and in the second stage re-scores these hypotheses using more complex models such as extended bigrams, as shown in Figure 3. The one with the highest sentence probability is then hypothesized as the conversion result. The search algorithm for generating N-best sentence hypotheses using word bigram from the word lattice is briefly summarized in the following steps :

1. Perform forward dynamic programming algorithm and compute α_w using equation (7) for every word w in the word lattice
2. Push initial hypothesis which contains sentence end node only into the stack
3. Pop the best hypothesis h from the stack. Hypothesis h is started with word w
4. Go To 9 if h is a complete sentence and N-best sentences have been generated
5. Go To 3 if h a complete sentence and N-best sentences have not been generated
6. For each word v precedes w, extend hypothesis h to word v, create a new hypothesis h', and compute the score for h'

   $$\beta(h') = \beta(h) + \log \Pr(w | v) \quad (10)$$

$$score(h') = \alpha_v + \beta(h') \tag{11}$$

7. Push all the new hypotheses *h'* into the stack
8. Go to step 3
9. Return the N-best sentence hypotheses

To ensure the global optimum is included in the N-best sentence hypotheses, large N is usually required. In our experiments , we observe N=200 is sufficient enough for our test data, and N larger than 200 does not yield higher accuracy. The final system with extended bigram results in 95.3% conversion accuracy, representing a 11% error reduction as compared with word bigram only.

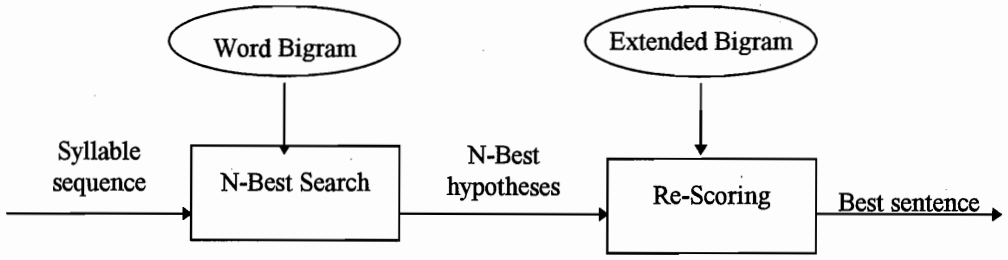


Figure 3. N-Best Search paradigm for the phoneme-to-text conversion using extended bigrams

4. Merged Long-distance Bigram and Stack Decoder

In the previous section we successfully integrate the extended bigrams into our phoneme-to-text conversion system. However, the proposed model and algorithm are not practical for real applications. The drawbacks are twofold. Firstly, it requires *d* bigram tables to store the long-distance information for extended bigrams with window size *d*. In our case *d* equals to 5, which means the storage requirement is increased by 400% but only achieves 11% error reduction. Secondly, for the N-best search algorithm, it takes a lot of efforts to generate many hypotheses and re-scores them, resulting a very inefficient system. On a Pentium-Pro 200 MHz machine, the system can only convert 22 Chinese characters per second.

To reduce the storage requirement, we merge the extended bigram tables from distance 2 to *d* to form a single long-distance bigram, diminishing the influence brought by different distances. More precisely, we assume the current word w_i, in addition to the its immediate previous word w_{i-1}, can also be predicted by any word in $W_{i-2,i-d} = w_{i-2}w_{i-3}...w_{i-d}$ with equal

possibility. In our model, the probability of word w_i given its past word sequence $W_{i-1,i-d} = w_{i-1}w_{i-2}...w_{i-d}$ is

$$\Pr(w_i | W_{i-1,i-d}) \cong (1-\lambda_{ld}) \Pr(w_i | w_{i-1}) + \lambda_{ld} \Pr_{ld}(w_i | W_{i-2,i-d}) \quad (12)$$

where $\Pr(\bullet|\bullet)$ is the word bigram probability, $\Pr_{ld}(\bullet|\bullet)$ the long-distance bigram probability. The long-distance model weight λ_{ld} can, again, be optimized using equation (9). For simplicity in the search algorithm, $\Pr_{ld}(\bullet|\bullet)$ is approximated by

$$\Pr_{ld}(w_i | W_{i-2,i-d}) \cong \max_{j=i-2}^{i-d} \Pr_{ld}(w_i | w_j) \quad (13)$$

Here we assume the long-distance dependency relies on the relations of word pairs. To avoid over-smoothing the parameters by merging many bigram tables, we select only those word pairs having high mutual information within a window based on the approaches proposed by Church and Hanks in (Church 1988). This concept is similar to that of Trigger Pair language model using Maximal Entropy Approach (Rosenfeld 1996). The differences are that we are dealing with long-distance relations at sentence level instead of document level in (Rosenfeld 1996) for adaptation, and using conventional Maximum Likelihood Approach to estimate our long-distance model parameters.

To integrated this merged model, we employ a stack decoder which contains a forward dynamic programming pass and a backward stack decoding pass. Unlike N-best search in the previous section, the long-distance probability are integrated earlier in the forward pass, so as to provide a more accurate heuristics for the backward stack decoder. When performing the forward dynamic programming, each node will examine all its preceding, but not adjacent, nodes in the lattice for long-distance bigram pairs, pick the maximal one, and integrate it into the computation of α. The corresponding equation for computing α_w for word w is as follows :

$$\alpha_w = (1-\lambda_{ld}) \times \max_u [\alpha_u + \log(w|u)] + \lambda_{ld} \times \max_v \log \Pr_{ld}(w|v) \quad (14)$$

where u are words preceding and adjacent to w and v are words preceding but not adjacent to w in the lattice. This step seems very time-consuming, since for every node we need to examine all its preceding nodes in the lattice. Fortunately, because the number of long-distance bigram pairs

are quite limited after the selection based on mutual information, we can use a very simple bookkeeping technique to efficiently identify all possible pairs having long-distance relations in the lattice.

In the backward pass, the stack decoder use these α in each nodes for heuristic functions. Since the computation of α in equation (14) are always over-estimated, the stack decoder is A* admissible, and the first decoded sentence is guaranteed to be the global optimum. The search algorithm is summarized in the following steps :

1. Perform dynamic programming algorithm and compute α using equation (14)
2. Push initial hypothesis containing only sentence end into the stack
3. Pop the best hypothesis *h* from stack. Hypothesis *h* is started with word *w*
4. If *h* is a complete path, go to step 8
5. For each word *v* immediately precedes *w*, extend hypothesis *h* to word *v*, create a new hypothesis *h'*, and compute $\beta(h')$ using equation (1), (12) and (13)

   $$score(h') = \alpha_v + \beta(h') \qquad (15)$$

6. Push all the new hypotheses *h'* into the stack
7. Go to step 3
8. Return the decoded sentence

In our system, we use DEAP data structure (Horowitz 1991) to implement the stack, which gives O(logN) complexity for the push and pop operations. The stack size is limited, and hypotheses ranked lower than this limit is truncated permanently. Experiments show that the accuracy is almost the same as that of extended bigram and N-best search, while both storage requirement and performance are significantly improved. On a Pentium-Pro 200 MHz machine, this system can convert about 60 Chinese characters per second in average.

5. Experimental Results

In our experiments, the language model training data are all newspapers provided by CKIP of Academia Sinica, containing about 12 million words. The lexicon contains about 42K words in which all Chinese characters are included as single-character words. Before training the language model, sentences from the training data are first segmented into word sequences in a data-driven, iterative manner. The algorithm is initialized by using a longest match strategy, and at each iteration aligns sentences against word entries in the lexicon based on bigram criterion,

aiming to reduce the perplexity for the training data. After several iterations , the lowest perplexity we obtained for these training data is 156. The test data contains about 110K Chinese characters, 80% of them are from newspaper, and the rest 20% are from other domains including prose, tourism, sports, art and ecology, etc. The perplexity for the test data is 382. Using our baseline system with word bigram only, we obtain 94.7% character accuracy for the test data, and the conversion speed is about 126 Chinese characters per second.

For extended bigrams, we use d=5 and assume the effect of long-distance dependency can be ignored for distance greater than 5. After optimized by deleted interpolation algorithm, the mixture weight corresponding to each distance is shown in the following table :

λ_1	λ_2	λ_3	λ_4	λ_5
0.68	0.16	0.08	0.03	0.05

Table 1. Mixture weight corresponding to different distances

In this table we can observe the weight for d=1 takes the lion's share, indicating a large portion of context dependency have been modeled by bigram. The mixture weight for d=2 is decent. Modeling context dependency for distance longer than 1 should be beneficial. The experimental results using extended bigrams and N-best search are shown in Table 2, where the accuracy and conversion speed with respect to different numbers of hypotheses are listed. In this table, the accuracy is saturated when the number of hypotheses exceeds 200. The best result obtained is 95.3%, representing a 11% error reduction as compared with 94.7% of the baseline system. The speed for this system is about 22 characters per second.

N	1	10	20	30	50	100	**200**	300	400
Accuracy (%)	94.7	94.8	95.1	95.1	95.1	95.2	**95.3**	95.3	95.3
Speed (char./sec)	126	112	94	75	49	38	**22**	12	8

Table 2. Experimental results for the extended bigrams : Character accuracy and conversion speed with respect to different N in N-best search

For the merged long-distance bigram, we truncate long-distance pairs based on mutual information criterion. Using stack decoder with stack size 16K, the accuracy obtained given different numbers of parameters are shown in Table 3.

Number of long-distance pairs	10K	20K	50K	**100K**	200K	400K	800K
Accuracy (%)	94.9	94.9	95.1	**95.3**	95.3	95.2	94.8

Table 3. Experimental results for the merged long-distance bigram : Character accuracy with respect to different numbers of long-distance bigram pairs

It can be found that the accuracy is increased as the number of long-distance pairs proceeds from 10K to 100K, but is degraded when more than 200K long-distance pairs are included in the language model. Merging these extended bigram tables can result in inaccurate parameter estimation, and therefore introduce some noise information. However, truncation based on mutual information helps to remove these noisy information, and turns out to be a good industrial tradeoff for the consideration of efficiency. In this experiment, the best result obtained is 95.3%, which is the same as that of extended bigram and N-best search. The search efficiency, however, is significantly improved. We achieve 60 characters per second for the conversion speed.

6. Conclusion

Modeling long-distance dependency reduces the error rate reasonably for the Chinese phoneme-to-text conversion task. In this paper, we present two models for the long-distance context dependency, extended bigrams and merged long-distance bigram. Two approaches are also presented for the integration of these long-distance models, N-best search and stack decoder. Tree-trellis based N-best search can integrate extended bigram language models, but need to generate many sentence hypotheses to cover the global optimum. Besides, huge storage is required for the extended bigram models. In order to reduce the storage requirement, we merge all the extended bigram tables from distance 2 to d to form a single long-distance bigram table, and derive a very efficient stack decoder for the integration of this merged long-distance model. Experiments show that the integration of long-distance information reduces error rate by 11%. Using stack decoder and merged long-distance bigrams, the system can convert 60 Chinese characters per second.

Reference

Church, K. W. and Hanks, P. "Word Association Norms, Mutual Information, and Lexicography", ACL 1988, pp. 76-83

Duda, R. O. and Hart, P. E. "Pattern Classification and Scene Analysis" Wiley, New York, 1973

Horowitz and Sahni, "Fundamentals of Data Structure in PASCAL" third edition. Computer Science Press, pp. 534-541, 1991

Hsu, W. L., "Chinese Parsing in a Phoneme-to-Character Conversion System Based on Semantic Pattern Matching", Computer Processing of Chinese and Oriental Languages, Vol. 8, No. 2, December 1994, pp. 227-236

Huang, X. D. et al, "An Overview of the SPHINX-II Speech Recognition System", Proceeding of the ARPA Human Language Technology Workshop 1993. Published as *Human Language Technology*, pp. 81-86, Morgan Kaufmann

Kuo, J. J. "Phonetic-Input-to-Character Conversion System for Chinese Using Syntactic Connection Table and Semantic Distance", ICCPOL 1995, pp. 286-292

Ney, H. et al, "On Structuring Probabilistic Dependences in Stochastic Language Modeling". Computer Speech and Language (1994) 8, pp. 1-38

Placeway, P. et al, "The Estimation of Powerful Language Models from Small and Large Corpora", ICASSP 1993, pp. II-33 to II-36

Rosenfeld, R., "A Maximum Entropy Approach to Adaptive Statistical Language Modeling", Computer Speech and Language (1996) 10. pp.187-228

Soong F. and Huang, E. F. "A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition:, ICASSP 1991, pp. 705-708

Wang, H. M. et al, "Complete Recognition of Continuous Mandarin Speech for Chinese Language with Very Large Vocabulary Using Limited Training Data", IEEE trans. On Speech and Audio Processing, vol. 5, NO. 2, March 1997

Automatic Speaker Identification Based on Fuzzy Theory and Neural network Using Genetic Algorithm

Ching-Tang Hsieh, Eugene Lai and You-Chuang Wang
Department of Electrical Engineering,
Tamkang University, Taipei , Taiwan,
Republic of China

Abstract

This paper proposes a two-stage speaker identification structure, using the average value of first formant (Vi) and zero crossing rate as the parameters and then preliminarily clustering the speech data by the distributed fuzzy rules to eliminate unnecessary silence and consonants. In order to fasten the operation speed to achieve the real time system, we also use the genetic algorithm to screen out unnecessary fuzzy rules. The results of the experiment show that the distributed fuzzy rules do effectively cluster the data and have the adaptability to independent speakers. Also, in screening the fuzzy rules, the genetic algorithm can greatly eliminate unnecessary fuzzy rules, and the difference of the recognition rate is under 1%. After preliminarily screening the speech data, we use the back-propagation neural network as the last speaker recognition structure. Since the system has eliminated silence and less stable consonants, we find that, according to the results of the experiment, the whole recognition rate can also get well-improved. Furthermore, this two-stage recognition structure proposed in the paper makes speaker identification automatic.

1. Introdution:

Generally, speaker recognition can be divided into two parts: speaker identification and speaker verification. Speaker verification refers to whether the speech samples belong to some specific speaker or not. Thus, the result can only be yes or no and it is caculated by the critical value. Of course, this critical value needs to be acquired from the experiment or set by the experts. The setting of the critical value affects the recognition rate of the whole system, and there are many studies in this aspect. Speaker identification system compares the speech samples with referential samples of all the speakers in the data base and finds out the fittest refer-

ential sample. This sample then belongs to the speaker. Finally, using the statistcs or induction, the system can acquire the final result. In comparison of speaker identification system with speaker verification system, the main difference is that speaker identification system has N possible choices but not just either one or the other, and with the increase of speakers, the system becomes more complex. Therefore, the misjudgement rate will increase a little. The operation style of both "speaker verification" and "speaker identification" can be divided into text-dependent and text-independent. "Text-dependent" means the text used in the training system is the same as that the test system uses. This is simpler to the system. On the contrary, "text-independent" means that there is no limitation for the text used in the test system. This style, of course, is more complex to the system. And in comparison with text-dependent, the misjudgement rate of text-independent is higher. Although many scholars have investigated in speaker identification and have a good result on recognition [1]-[4], they lack an integrated structure. Thus, this paper proposes a two-stage recognition structure, using the average value of first formant (Vi) [7] and zero crossing rate as parameters and then preliminarily clustering the speech data by the distributed fuzzy rules to eliminate unnecessary silence and consonants. In order to fasten the operation speed to achieve the real time system, we also use the genetic algorithm to screen out unnecessary fuzzy rules. The results of the experiment show that the distributed fuzzy rules do effectively cluster the speech data. The genetic algorithm can almost screen the fuzzy rules to 1/4 of its original number, and the difference of recognition rate is under 1%. As to the recognition structure, we use the back-propagation neural network, which has the highest accuracy, to do the last identification of the speakers. Since the system has eliminated the silence and less stable consonants of contineous speech, we find that, from the results of the experiment, the whole recognition rate of the system can also get well-improved. This two-stage recognition structure proposed in the paper makes speaker identification automatic.

The contents of the following sections are: Section II introduces the application of the distributed fuzzy rules and the genetic algorithm in the preliminary classification of the speech data; Section III introduces the back-propagation neural network and the speaker recognition structure mentioned in this paper; Section IV is the result and evaluation of the experiments and Section V is the conclusion.

2. The preliminary classification of the speech

The preliminary classification of the speech data is very important to a good speaker identification system. We will use Vi and zero crossing rate as the parameters and then use the distributed fuzzy rules to cluster the characters of the speech data. In order to make the system optimal, we also use the genetic algorithm to screen the fuzzy rules to make the system reach the goal of speedy operation and achieve the real time system.

2.1 Distributed fuzzy rules

Hisao, Ken, and Hideo[5] proposed the "Distributed Fuzzy Rules" to cluster the numerical data using the triangular membership function. For only 3 classes and 9 training samples, the correct ratio for clustering unknown samples is up to 90%. There are two conclusions: (1) Under the same fuzzy partition, the correct ratio by the distributed fuzzy rules is higher than that by the ordinary fuzzy rules; (2) Even for fewer training samples, the correct ratio by the distributed fuzzy rules to cluster unknown samples is still higher. These properties are beneficial for clustering the features of large speech data without many training data. The common by used types of the membership function of the fuzzy rules are triangular membership function, exponential membership function, Mexico hat membership function, and so on. Exponential membership function is shown below:

$$\mu_i^k(x) = \exp(-\beta^2(x - a_i^k)^2) \qquad i = 1,2,.....,k \ (k \geq 2) \qquad (1)$$

where μ_i^k is the membership function of subspace A_i^k and

$$a_i^k = (i-1)/(k-1) \qquad i = 1,2,.....,k \qquad (2)$$

The ordinary fuzzy rules can be described as

If x is X_i^L and y is Y_j^L
then [x,y] belongs to $[X_i^L, Y_j^L]$
i , j = 1,2,.....,L (3)

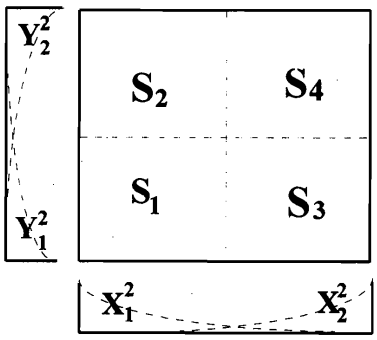
where [x , y] is one sample in unit square [0,1]x[0,1] ,and $[X_i^L, Y_j^L]$ is the subspace of unit square, and L is the fuzzy partitions. Making some modifications of eq. (3) as

If x is X_i^k and y is Y_j^k
then [x,y] belongs to $[X_i^k, Y_j^k]$
i, j =1,2,....,k; k = 2,.....,L (4)

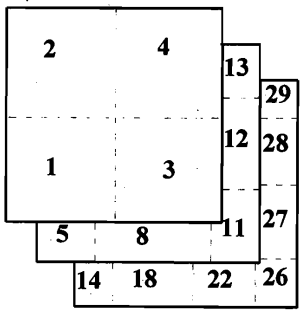
becomes the distributed fuzzy rules. Different number of fuzzy partitions is the difference between the ordinary fuzzy rules and the distributed fuzzy rules. For L partitions, the distributed fuzzy rules are operated on L-1 unit spaces, i. e. from 2 partitions to L partitions. The ordinary fuzzy rules are operated only on one unit space, i. e. the unit space with L partitions. Fig. 1 shows the representations of different fuzzy rules. Clustering every subspaces from the training data to obtain eq. (5) is the training purpose.

If x is X_i^k and y is Y_j^k
then [x,y] belongs to class-x (5)

In eq(5), class-x is the class of own data; the class with maximum membership among all subspace classes is the class of unknown data. More details can be seen from reference[5].



(a) Labels and indices of fuzzy if-then rules (L=2)



(b) Distributed Fuzzy Rules (L=4)

Fig 1. The representations of different fuzzy rules

2.2 The application of genetic algorithm in fuzzy rules

We can get the set of fuzzy rules by training some speech data. Our problem in this section is to select fuzzy rules from all the fuzzy rules to construct a compact rule set with high classification power. We can briefly describe the genetic algorithm operations as follows[6]:

i) Initialization: Generate some initial populations randomly that contain some string ($S=S_1S_2.....S_N$) where

N is the total number of fuzzy rules.

S_n=1 denotes that the nth rule is selected.

S_n=-1 denotes that the nth rule is not selected.

ii) Fitness: Our purpose is to maximize the number of correctly classified speech data by the selected fuzzy rules set S and to minimize the number of fuzzy rules in S. So, the fitness value of each string can be formulated as

$$F(s) = W_{NCP} * NCP(s) - W_S * |S| \tag{6}$$

where $F(s)$ is the fitness value($F(s) \geq 0$), NCP(s) is the number of correctly classified speech data by S and |S| is the number of fuzzy rules in S. $W_{NCP}$and W_s are positive weights.

iii) Reproduction: The selection probability of the individual S in new generation is proportional to its fitness value.

$$P(s) = \frac{f(s)-f_{\min}(\Psi)}{\sum_{s' \in \Psi} \{f(s')-f_{\min}(\Psi)\}} \tag{7}$$

where $f_{\min}(\Psi) = \min\{f(s) : S \in \Psi\}$

iv) Crossover: We apply one point for crossover to the pair of selected individuals such that we can get new strings.(The step repeats p/2 times, where p is the number of populations).

v) Mutation: This operation can prevent strings to locate into local optimization.

$$S_n \rightarrow S_n * (-1) \tag{8}$$

Repeat steps iii, iv, and v until satisfy the stopping condition.

3. Speaker identification structure of the back-propagation neural network

The back-propagation neural network used in this paper is the most representative and common neural network nowadays. The basic rule of the back-propagation neural network is to use the concept of the gradient steepest decent method to minimize the error function. Because of the practicality and high recognition rate of the back-propagation neural network, this paper applies it to be the main recognition structure of speaker identification.

3.1 The application of genetic algorithm in back-propagation neural network

Although the back-propagation neural network has satisfying results, including high learning accuracy, fast recollection speed, etc.., it still has some unavoidable defects. Many scholars are making researches in this aspect.

Local minimum is the most troublesome problem in the defects of the back-propagation neural network. Because the back-propagation neural network is based on the gradient steepest descent method, it will unavoidably be puzzled by the local minimum. Even in the process of minimizing the error function, the weighting of the network falls into a local minimum of the error function and can't jump out, so that the convergence is incomplete and the error function does not reach the global minimum. This phenomenon is caused by two major reasons: (1) the order of the training samples: Because the adjustment of the weighting of the networks adopts single pattern learning, that is, the weighting adjusts one time in each training sample, the order of the samples will influence the learning result. Fortunately, from the experiment, we know the order of the samples doesn't influence much on the last training result of the system. This paper will not consider this factor. (2) the initial value of the weighting of the networks: the initial value of the weighting will influence the efficiency of the whole system, and if the value is good, the system will approach the optimum quickly; on the contrary, if the value is bad, what the system will get is the local minimum.

Because the genetic algorithm uses the multiple points search, it has a good effect on the optimum of the system. Therefore, this paper uses the advantages of the genetic algorithm to improve the problem of the local minimum that the back-propagation neural network faces. Of course, the algorithm can't fully prevent the network from the trouble of the local minimum. But, it can make the network reach the multiple points search to avoid the result of local minimum as much as possible. Also, to increase the search ability of the system, this paper randomly adds a perturbation to the process of searching. According to the experiment, we find

that adding the random perturbation really helps the search of the system. The way of perturbation is listed below:

$$W^{n+1} = W^n + \Delta W + \xi \tag{9}$$

The whole process of applying the optimum of the genetic algorithm to improve the back-propagation neural network is described below:

(i) Initial population: Randomly giving the weighting on some different groups and regarding them as the initial population in the whole operation.

(ii) Fitness function: Because the training of the back-propagation neural network is to acquire the minimum of error function, the fitness function of every group can be identified as the total error of the neural network.

(iii) Acquiring the weightings of every group by Error Back-propagation. Then figuring out the total error of every group.

(iv) Mutation: Randomly choose one group from the population. Then, randomly assign the position of the selected group and change the weighting value to produce a new group. The mutation times are based on the fitness of the group. That is, the group which has large fitness value has more mutation times; on the contrary, the group which has small fitness value has an obvious declination of its mutation times. This process needs to be done once to all groups.

(v) Cross-over: Randomly choose two groups from the population. Then, use the cross-over method to exchange part of the weighting value to produce new groups. This process is usually executed P/2 times (P is the group number of the population).

Repeat step (iii), (iv), and (v) until the fitness value fits the requirement of the system.

3.2 Speaker identification structure

In the way of speaker identification, because the change of the speech of text independent is too complex, it is necessary to preliminarily cluster and screen the speech data. On the other hand, vowels hold the most part in Chinese text and are more stable, so this paper uses zero-crossing rate and Vi as the parameters of preliminary classification to eliminate unnecessary silence and consonants. In the preliminary classification, this paper uses the distributed fuzzy rules to screen the vowels, because it can get quite good result with very few training samples.

In the last recognition, we use the highly accurate back-propagation neural network to be the last speaker recognition structure. The whole recognition process is represented in Fig. 2.

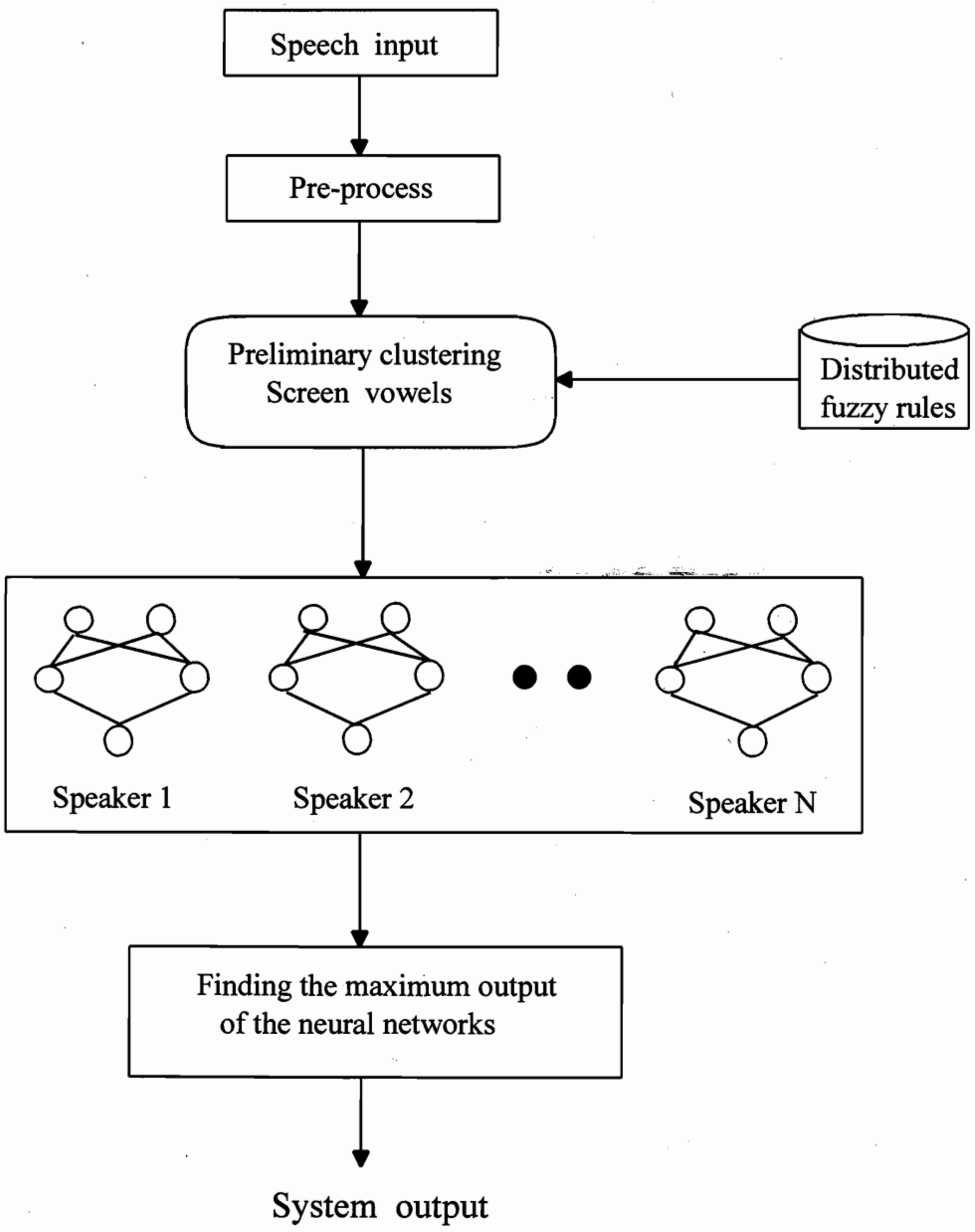


Fig. 2 The recognition structure of two-stage speaker identification

4. Experimental results and evaluation

This section will describe an experiment in the two-stage recognition structure, including the preliminary classification and the screening of the speech data in the first part and the speaker identification structure of neural network in the other part.

4.1 The experiment of the preliminary classification of the speech data

The speech data in this experiment include seven blocks for each of two males and two females. In each block, they read any article for four seconds. The sampling rate is 10kHz. Each analysis frame is 25.6 ms and the overlap between analysis frames is 15.6 ms. The training data come from the first five blocks and the test data come from the last two. Table I shows the classification of speech phonemes. Table II shows the correct recognition rate of classification from partition number L=6 to L=15. From Table II, we can find that the recognition rate in different fuzzy partition has different results for different classes. But, on the whole, the average correct recognition rate increases with the partition number, and the correct recognition rate of each class gradually converges. However, though the larger fuzzy partition number helps to enhance the average correct recognition rate, the fuzzy rules increases tremendously with the increase of partition number (the distributed fuzzy rules are sum of square of the fuzzy partition; if fuzzy partition number is L, then the distributed fuzzy rules $S_{ALL} = 2^2 + 3^2 + ... + L^2$). So, in considering how to increase the recognition rate, we should also think about the execution efficiency of the system. In order to test the distributed fuzzy rules to see if it can still have good result with fewer training data, we use different blocks to be the training data. Table III shows the results. Fig 3 and Fig 4 are the classified diagrams from one training and five training blocks (0 ,1 and 2 denote silence, consonants and vowels, respectively). From table III, we find that the whole classification rate does not increase with the growth of the training data. Thus the distributed fuzzy rules really can have good classification result with very few data.

In order to understand the relation between the adjustable coefficient of membership function and the result of classification, we use 3 blocks as the training data and the fuzzy partition number L=6. Table IV shows the results of the experiment. We find that the rates for silence and vowels decline and the rates for consonants increases with the increase of adjustable coefficient; and the whole classification rate almost doesn't change. Thus, we may know that the setting of adjustable coefficient affects the recognition rate of each class.

Table I. Relationship between Classes and Phonemes

Class	Phoneme
Silence	silence
Consonants	f,d,t,g,k,h,j,b,p,y,m,ch,sh,tz,ts,s,r,l
Vowels	i,u,a,o,e,io,ai,ei,au,ou,el,ia,ie,iau,iou,ua, uo,uai,uei,ue,iue,iua,iu,n,ng

Table II. Classification results with different values of L by distributed fuzzy rules using triangular membership function.

	L=6	L=12	L=13	L=14	L=15
Silence	94.35%	84.85%	84.85%	84.77%	85.21%
Consonants	70.11%	83.54%	83.77%	83.84%	84.14%
Vowels	98.76%	96.02%	95.95%	95.91%	95.87%
average	87.74%	88.14%	88.19%	88.17%	88.41%
fuzzy rules	90	649	818	1014	1239

Table.III The classification result with different training blocks (L=6, β=6)

Training Blocks	1	2	3	4	5
Silence	86.9%	89%	90%	90%	92.1%
Consonants	72.2%	73%	72.8%	71.3%	69.6%
Vowels	99.3%	99%	99%	98.9%	99%
Average	89.9%	90.2%	90.3%	89.8%	89.7%

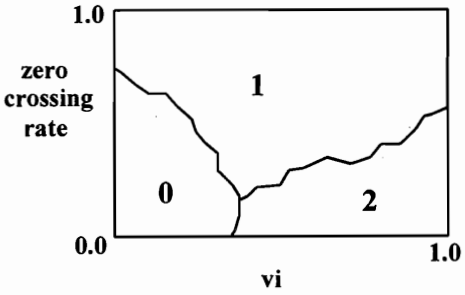


Fig 3. The distribution of classes for 1 training block

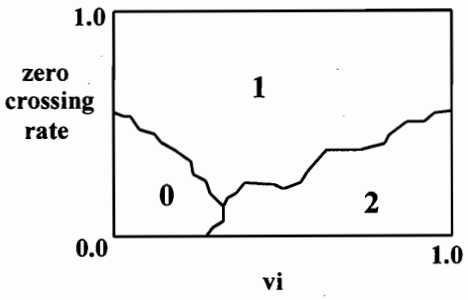


Fig 4. The distribution of classes for 5 training block

Table IV. Classification results with different values of β by distributed fuzzy rules using exponential membership function(L = 6).

	β = 6	β = 7	β = 8	β = 9	β = 10
Silence	88.70%	88.24%	87.77%	87.38%	87.23%
Consonants	79.75%	82.04%	82.96%	83.40%	83.70%
Vowels	98.05%	95.66%	95.30%	94.90%	94.68%
average	88.83%	88.65%	88.68%	88.56%	88.54%

Next, we observe the effect of the genetic algorithm used in this paper. When the fuzzy partition number L changes, we use the genetic algorithm to screen the fuzzy rules. Table V and Table VI are the results of the experiment. From the Tables, we find that although fuzzy rules have been greatly eliminated, the whole classification rate doesn't have obvious declination. We also find that using the exponential membership function to do the classification is better than using the triangular membership function. The number of average fuzzy rules can decline to 1/4 of its original amount. Thus, using the genetic algorithm to screen the fuzzy rules really works. Fig 5 shows the diagram of the distribution of classes after being operated by the genetic algorithm. There is no obvious difference between the distribution of each class before and after the operation of the genetic algorithm.

Table V. Classification results with different values of L by genetic algorithm (Using triangular membership function).

	L=6	L=12	L=13	L=14	L=15
Silence	89.93%	81.13%	83.44%	87.87%	83.52%
Consonants	78.63%	85.17%	85.84%	81.25%	85.47%
Vowels	96.63%	96.79%	95.20%	96.89%	95.14%
average	88.40%	87.70%	88.16%	88.67%	88.04%
fuzzy rules	38	239	314	384	485

Table VI. Classification results by genetic algorithm. (Using exponential membership function)

	Silence	Consonant	Vowel	fuzzy rules
1	83.13	85.62	94.19	20
2	90	80.43	96.01	21
3	87.15	82.28	95.48	15
average	86.76	82.78	95.23	18.7

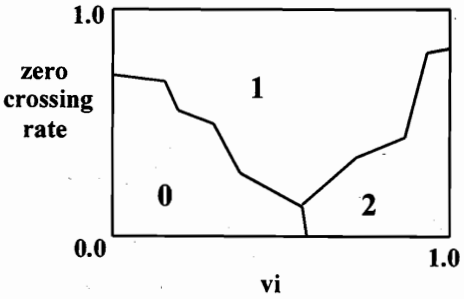


Fig 5. The distribution of classes with GA for 3 training blocks (exponential membership)

4.2 Performance evaluation

This evaluation of speaker identification experiment was conducted in the following manner. First, the test speech was produced a sequence of feature vectors $\{x_1, x_2, ..., x_t\}$. To evaluate different test utterance lengths, the sequence of the feature vectors was divided into overlapping segments of T feature vectors. The first two segments from a sequence would be

$$\underbrace{x_1, x_2, \ldots\ldots\ldots, x_T}, x_{T+1}, x_{T+2}, \ldots\ldots$$

Segment 1

$$x_1, \underbrace{x_2, \ldots\ldots\ldots, x_T, x_{T+1}}, x_{T+2}, \ldots\ldots$$

Segment 2

A test segment length of 5 seconds corresponds to T=250 feature vectors at a 20 ms frame rate. Each segment of T vectors was treated as a separate test utterance.

The identified speaker of each segment was compared with the actual speaker of the test utterance and the number of segments correctly identified was tabulated. The above steps were repeated for test utterances from each speaker in the population. The final performance evaluation was then computed as the percent of correctly identified T-length segments over all test utterances

$$\%\ \text{correct identification} = \frac{\#correctly\text{–}identified\text{–}segments}{total\#ofsegments} \times 100 \qquad (10)$$

The evaluation was repeated for different values of T to evaluate performance with respect to test utterance length [1].

4.3 The experiment of two-stage speaker identification

In this section, the training data includes twenty-seconds articles and six groups of telephone numbers; the test data are the four records of sentences and listed in Table VII. A 25.6 ms Hamming window is applied to the speech every 10 ms, and the feature consists of the 10 cepstral coefficients (LPC) and the 10 dynamic spetral coefficients. Table VIII shows the experimental results only using the back-propagation neural network as the recognition structure.Table IX shows the experimental results after the preliminary disposition by the

distributed fuzzy rules. From the tables, we can see that we can eliminate unnecessary silence and less stable consonants in the continuous speech when the system is preliminarily classified and screened by the distributed fuzzy rules. Then, we may screen the vowels to do the networks training. The proposed system of this paper not only can fasten the operation speed but also can enhance the total recognition rate.

Table.VII The contents of utterance.

	Contents
1	高壓（gao ya）
2	曲棍球（qu gun qiu）
3	雙管齊下（shuang guan qi xia）
4	淡水捷運站（dan shui jie yun zhan）
5	但願我們能夠永遠在一起（dan yuan wo men neng gou yong yuan zai yi qi）

Table.VIII The experimental results of text-independent speaker identification.
(Using the back-propagation neural network)

Segment length (sec)	5 speakers	10 speakers	20 speakers
1	79.5%	75.2%	68.4%
2	85.9%	80.3%	76.2%
3	89.2%	84.7%	81.2%
4	93.5%	90.1%	85.1%

Table.IX The experimental results of text-independent speaker identification.
(Using the distributed fuzzy rules and the back-propagation neural network)

Segment length (sec)	5 speakers	10 speakers	20 speakers
1	84.7%	78.8%	75.1%
2	92.8%	84.3%	81.5%
3	97%	90.5%	87.2%
4	100%	93.1%	90.3%

5. Conclusion

In the preliminary disposition, speaker identification system mentioned in this paper uses the zero-crossing rate and Vi as the parameters, and then preliminarily clusters and screens the speech data by the distributed fuzzy rules and the genetic algorithm, in order to eliminate the unnecessary silence and consonants. Besides, the system has the adaptability to independent speakers. As to the recognition, in order to reduce the rate of falling into local minimum, we use the characteristics of the genetic algorithm to do multiple points search to the neural networks. From the results of the experiment, we can find that the proposed two-stage recognition structure not only improves the recognition rate but also makes speaker identification autumatic.

Reference

[1] Dougla A. Reynolds and Richard C. Rose, "Robust Text Independent Speaker Identification Using Gaussian Mixture Speaker Models," IEEE Trans on speech and Audio Processing, Vol. 3, No. 1, pp. 72-83, Jan. 1995.

[2] Kevin R. Farrell, Richard J. Mammone and Khaled T. Assaleh, "Speaker Recognition Using Neural network and Conventional Classifiers," IEEE Trans on Speech and Audio Processing, Vol. 2, No. 1, Part II, pp. 194-204, Jan. 1994.

[3] T. Matsui and S. Furui, "A Text-independent speaker recognition method robust against utterance variations," in Proc. IEEE ICASSP, 1991, pp. 377-380.

[4] S. Furui, "Comparison of speaker recognition methods using statistical features and dynamic features," IEEE Trans. Acoust., Speech, Signal Processing, Vol. ASSP-29, pp. 342-350, June 1981.

[5] Hisao Ishibuchi, Ken Nozaki and Tanaka, "Distributed representation of fuzzy rules and its application to classification," Fuzzy Sets and Systems 52(1992), pp. 21-31.

[6] Hisao Ishibuchi, Ken Nozaki, Naohisa Yamamoto and Hiedo Tanaka, "Construction of fuzzy classification systems with retangular fuzzy rules using genetic algorithms," Fuzzy Sets and Systems 65(1994), pp. 237-253.

[7] C. T. Hsieh and S. C. Chien, "Speech segmentation and clustering problem based on fuzzy rules and transition states," Twelfth IASTED Int. Conf. Applied Informatics, 1994.

A General Public Application of Pedagogic and Linguistic Vocations of Speech Synthesis: Ordictée

Marc Guyomard
Université de Rennes I, IRISA/ENSSAT 6, rue de Kerampont
BP 447 F-22305 Lannion Cedex France
guyomard@enssat.fr

Jacques Siroux
IRISA/IUT Lannion France

Dominique Pernici and Christophe Royer
ENSSAT Lannion France

Abstract

We present the Ordictée©software which allows the pupil to independently practice dictation exercises. Ordictée acts as a substitute for the teacher with regards to reading and dictation correction and allows the pupil to be almost completely in control in a non-stressful environment. Ordictée is made up of three modules which respectively allow the *pupil* to carry out the actual dictation, the *tutor* to administer a dictation database whilst adapting each text to the context of the dictation and the *designer* to generically specify the environment in which the dictations are carried out. Ordictée uses the *Proverb* software from the company "Elan Informatique".

1. General presentation

At the moment, we are witnessing an explosion in the software market of products aimed at assisting language learning. At the same time, research involving automatic speech processing (recognition and synthesis) which has been confined to the laboratories for a long time, is today ready for use. High quality products – especially in the domain of speech synthesis – are marketed. However, the merging of these trends remains often too superficial: the vocal terminals serve only as substitutes for the most classical means, without having a synergetic effect. On the other hand, some applications would be quite impossible without vocal based softwares. Such is the case of software designed to help with dictation. Nevertheless and especially for the French, such systems are rare (see however (Cotto 1990a, Cotto 1990b)) and those marketed are on this side of the state of the art.

In this context, we have decided to start the Ordictée project which enables a pupil to practice dictation exercises. Having selected a dictation which corresponds to their abilities, the pupil can ask Ordictée to read the dictation, to do the exercise (during which time they can type in the text being read to them). Following a second phase including further readings and subsequent corrections, the machine is asked to do the correcting.

In section 2 of the paper, we develop the structure of the software, demonstrate its various functions and explain its method of use. Section 3 presents the solution adopted to follow the typing activity. We then describe the algorithm employed for the correction

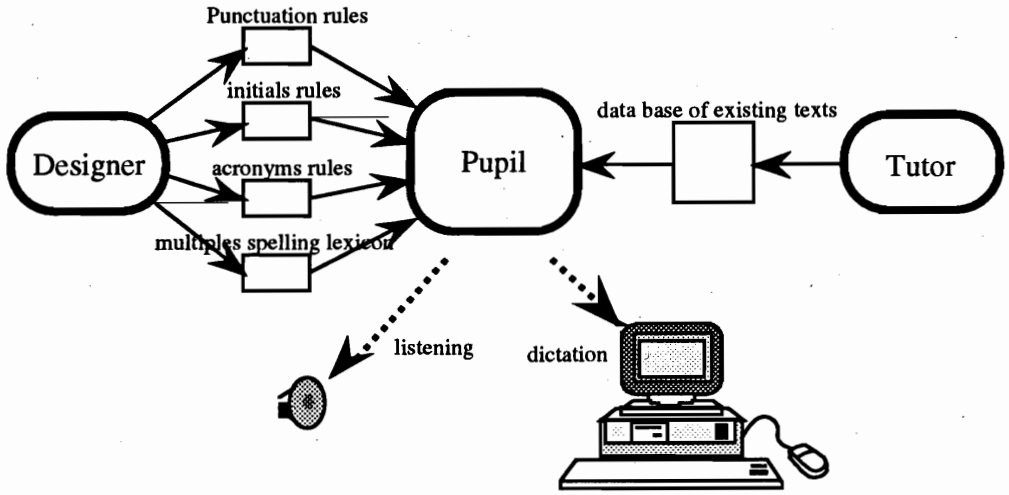


Figure 1: Architecture of the system

of dictations. An evaluation of the software, carried out with the pupils of a CM2[1] class, is then presented.

2. Structure and use of Ordictée

Ordictée is made up of three complementary modules: the *pupil* module, the *tutor* module and the *designer* module. The *pupil* module enables the pupil to carry out dictations. The *tutor* module aimed at the lecturer, allows them to form a collection of texts, adapting them in order to dictate them, whereas the *designer* module, aimed at the software manager, allows the management of lexicons and rule bases, giving rise to customised dictations and corrections (see figure 1).

The pupil module. The purpose of the pupil module is to act as a teacher for the different stages involved in dictation: reading, actual dictation and correction. Provided the tutor has designed the dictation in the correct way, Ordictée can (on request) offer linguistic explanations concerning the errors that are made. An average session with the pupil unfolds as follows: the pupil alters (should they wish to do so) the listening parameters (diction speed, pitch and volume of the voice).

They then select a dictation, listen to it at will, whilst looking at pictures illustrating the purpose of the text which reads.

They can then commence the exercise. Ordictée adapts itself to the pupil's rhythm, repeating the current phrase as much as necessary. The pupil can hear the dictation once more and carry out the necessary corrections. They then ask Ordictée to proceed with correction. Here, the various errors are revealed to the pupil.

The French vocabulary contains a group of about fifty words, which have, *out of context,* and for a given pronunciation and a given meaning, several correct spellings (in general, an advised spelling and one or several variants, such is the case with words like (clé, clef), (khôl, koheul, kohol)).

During the correction stage, and having spotted a fault, the pupil module consults

[1]Cours moyen de seconde année (the last year in French primary school).

a dictionary containing the various spellings for each word (see *designer* module), so as to check that the word thought to be incorrectly spelt does not have an alternative spelling.

The tutor module. The tutor module enables the lecturer to manage the dictations database. It also offers the option of going into the text in question in order to set markers informing the pupil module of particular pronunciations (initials, acronyms and heterophone words) or to section off the text into phrases to be repeated during the exercise.

The designer module. The designer module allows the management of a base of rules which are applied before the text is read. Three types of rules are permitted. They concern:

- initials,
- acronyms,
- punctuation.

During normal reading by an individual, or by a machine, punctuation is used to create prosody. Whilst reading the dictation, the punctuation must also be pronounced. The punctuation rules allow us to specify the pronunciation of each sign (?, !, "to the line", etc.).

The designer module also allows us to manage a dictionary of words, each of which has various spellings. This dictionary is used by the pupil module during correction.

3. Following the typing process

The objective of following the typing process is to assure synchronisation between the reading of a piece of dictation with the writing of a pupil. A second objective is to disrupt the pupil as little as possible, notably by avoiding pointless repetitions. Whereas in the traditional dictation, a range of information is available for the teacher to decide if the pupil has reached to the end of the typing, Ordictée is based uniquely on the information typed, on the speed the pupil types at and on the expected information. In a first version we founded a solution on an approximation of the number of characters given for deciding either to repeat the current piece of text or to proceed. This solution has rapidly shown its limits and we have implemented a solution of a heuristic nature which is based on hypotheses on the pupils' performance and on various measures performed during the typing. This solution is presented below.

3.1. Hypothesis on pupil performance

Following of the typing is based on the assumption that the pupils performance complies with the three following hypotheses:

1. The pupil stops typing
    - either because they think they have finished to type the current piece of text,
    - or because they do not know what they should write.

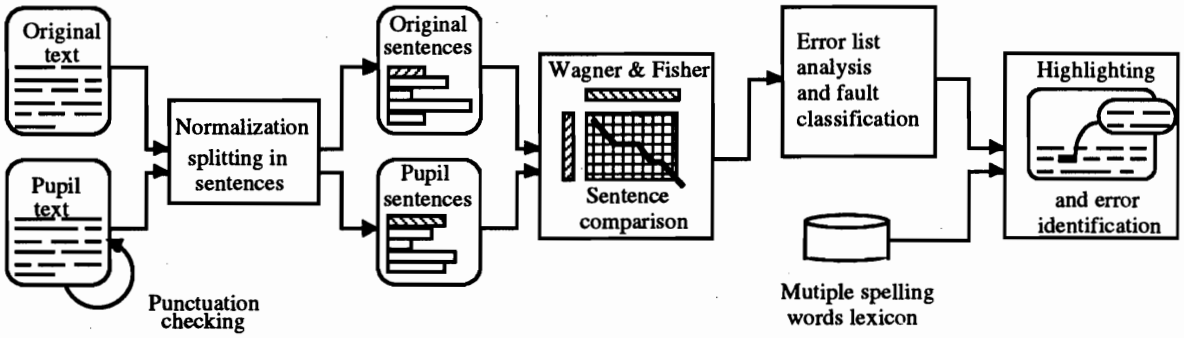


Figure 2: The various stages during the correction processing

2. The pupil who types knows what they should type. As the pupil types their text, it is therefore pointless to repeat the current piece.

3. A pupil can make a few mistakes but the insertion or forgetting of letters in a word is more probable than inserting or forgetting words. However it is not possible to found the following of typing uniquely on the counting of words.

3.2. The principal solution

Two decisions should be taken by Ordictée: "When should a piece of text be changed over?" and "When should the current piece be repeated?". The evaluation of these two conditions are based on the calculations of the following elements:

- the average speed of typing,
- the instantaneous speed of typing (which permits one to decide the inactivity of the pupil),
- where the pupil has not given up,
- the number of words remaining to be typed,
- where an audio output is still running.

The decision concerning the inactivity is based on the comparison between the speed of instantaneous typing and the average speed. The decision to withdraw is founded as regards to the inactivity of the pupil during several repetitions. The changing of the piece is then decided when the following condition occurs:

(enough words typed **and** inactive) **or** withdrawn)
and not audio output.

The decision of repetition is taken when the following occurs:

not enough words typed **and** inactive
and the audio output is finished
due to the timing (function of the average speed).

In a following version we envisage to enrich the solution presented below through using a comparison of typed text and of expected text through using a dynamic programming method (Gries 1988).

4. Spelling correction

When a dictation has been heard by the pupil, they can then ask the software to correct the mistakes. Ordictée has at his disposal the original text supplied by the tutor as well as the text issuing from the pupil. The correction procedure involves comparing both texts and thus finding the mistakes made by the pupil. This comparison is carried out by the Wagner and Fisher algorithm. We will go on to describe the principle of this algorithm and explain the different stages of correction.

4.1. The principal of the correcting algorithm

The principal difficulty of the correcting algorithm resides in the fact that the Wagner and Fisher algorithm produces a result in terms of comparing characters, whilst Ordictée should provide a word comparison. Beyond the simple application of the Wagner and Fisher algorithm, it is a question of having an algorithm at one's disposal which can be capable of interpreting an editing list of characters (deleting, adding and replacing of characters) in an editing list of words (misspelling, deleting, adding and replacing of words as well as merging or forbidding cutting of words).

4.2. The correction algorithm

The Wagner and Fisher dynamic programming algorithm (Du 1992, Stephen 1992, Wagner 1983) constructs a distance matrix between two strings of characters in order to determine the longest common sub-sequence with a minimum cost. The matrix is filled using the following recurrence formula:

$$D_{i,j} = \min\left(D_{i-1,j} + w(x_i, \varepsilon), D_{i,j-1} + w(\varepsilon, y_j), D_{i-1,j-1} + w(x_i, y_j)\right)$$

$D_{i,j}$ represents the distance between string X, with length i and string Y, with length j. $w(a,b)$ is the cost of the substitution of character $'a'$, by character $'b'$. $w(a,\varepsilon)$ and $w(\varepsilon,a)$ are the respective costs associated with the deletion and insertion of character $'a'$.

The Wagner and Fisher algorithm seeks the shortest route (in terms of character substitution, insertion or deletion), which allows the string X to be transformed into the string Y and tells us which is the longest common sub-sequence. This sub-sequence holds little interest for us, but it is nevertheless interesting to scan the matrix along the optimal route, in order to locate faults. This enables us to make a list of index couples showing the differences between the pupil's text and the original. Having analysed this list, we are then able to distinguish between several types of mistakes, and to offer adequate explanations.

4.3. The correction stages

A dictation is broken down into a group of sentences which are delimited by the punctuation marks at the end ('.', '!', '?' or '...'). It is thus necessary, during correction, to compare the chunks which correspond to the same sentence so as to ensure that the explanations supplied are coherent.

The first stage involves checking the pupil's text for punctuation, so that their sentences complying with the original text. If the punctuation is found to differ in any

way from the original, the pupil has the opportunity to alter it. It is possible for them to hear the dictation once again, in order to correct their text.

During the second stage the pupil's text is normalised: this processing allows Ordictée to remove the leftover spacing characters and to adopt the same typographic conventions as the original version (for example, a comma is preceded and followed by a space). The pupil's text is split into sentences which will be individually compared to those in the original text.

The third stage makes it possible to find the eventual errors of the pupil. The complexity of the Wagner and Fisher algorithm is in $O(m.n)$, where m and n are the respective lengths of strings to be compared. The effective time for processing long sentences can be non-negligible. So, in order to limit it, the sentence is pre-processed: Ordictée retains the sub-sequence between the word preceding the first wrong word and the word following the last wrong word. Reducing the search space in that way generally brings about better results. These sub-sequences are placed in the comparison matrix and we obtain the list of the pairs of errors. Its analysis determines the type of error.

Finally, the last stage consists of informing the pupil of the fault locations and to provide them with an explanation. Thanks to the groupings of pairs, we can see easily which word is incorrect and its correct form. Highlighting the incorrect word in the pupil's text is a delicate matter, as the pupil has total freedom as far as the use of spacing of characters is concerned. However, thanks to the method employed to notice differences between the two texts, we can see exactly where the mistake is. At the same time, we can identify the kind of mistake. Figure 2 recaps on the stages that we have described above.

4.4. Categories of detected errores

The difficulty with the interpretation of results supplied by the Wagner and Fisher algorithm in the context of spelling correction is mostly due to the lack of semantic knowledge concerning the sentences. In actual fact, this algorithm is based on the notion of distance between two strings. If it helps to locate an error, it does not give enough information which permits the exact description of mistakes. In these conditions it is preferable to limit the number of categories and to find ways to make trustworthy decisions to class mistakes. At the moment the following categories are taken into account:

1. a word forgotten by the pupil,
2. a word which was not requested,
3. a word is badly spelt.

The detection of the first two categories ensures the synchronisation of the correction, even when the pupil has forgotten part of the sentence, or has typed a few words which have not been requested. The omission of a word is detected in view, within the correction matrix, of a series of character insertions in the pupil's sentence. These characters correspond to the word expected in the original sentence. Similarly, a word which was not requested is identified when a series of deletions occur.

The last category is very general, as it indicates just a spelling mistake. The error is highlighted and the correct spelling is shown to the pupil. When a word is classed as incorrectly spelt, a few simple tests are carried out to decide if the error is merely due

to the misuse of capitals and small letters, or if it is because we are facing a word which has several possible spellings.

5. Evaluation

In spite of the fact that over the past twenty years its importance has reduced, dictation remains a key exercise within the pedagogics of the French primary school education system. Moreover, it is increasingly common to find schools with a computer room, offering the pupils an opportunity to familiarise themselves with a computing environment, yet at the same time allowing them to do exercises which help them to learn the French language.

Given all these reasons, together with the fact that the computer scientist should not make a product commercially available without it being put to the test among its future users, we have decided to proceed with an evaluation of Ordictée based on the behaviour of pupils at a primary school. Below we will describe the protocol used, the main observations made and the synthesis of interviews resulting from the exercises.

5.1. Protocol

We have used a class of pupils aged 10-12 from Saint Roch school in Lannion. Prior to the experiment, each of the pupils from the class has had some experience with a computer during computing sessions (about half an hour per week on PC-DOS without a mouse). As far as Ordictée is concerned, several demonstrations, followed by tests carried out by some pupils in front of the whole class, took place a few weeks before the stated evaluation. Each session takes the following form: the location is in an isolated room, in front of an assistant and a secretary.

1. The assistant explains to the pupil how the program works and the role of each button.

2. There is a start up program, so that the pupil can get used to the buttons, the mouse and double function keys (some French accents) and the use of the keyboard.

3. A dictation (of about thirty words) is selected by the assistant and the pupil is left to complete the dictation at there own pace, whilst leaving the pupil the possibility of calling the assistant if necessary. However, for each error detected and presented by Ordictée, the assistant asks the pupil to explain their mistake.

4. A directed interview based on the evaluation of Ordictée ergonomics as well as on the quality of synthesis, diction and correction is finally carried out.

During the presentation of Ordictée to the whole class, we have noticed that unlike normal dictation, the pupil feels challenged by the machine: they try, using an iterative process of self-correction automatic correction, to produce a perfect text. Achieving a fault-free text is an objective embedded in point three of the protocol.

Let us note that, following the demonstrations, all the pupils in the class understand the concept of the software and volunteers are plentiful.

5.2. Observation during the sessions

Ten pupils, who according to the teacher, have unequal abilities, are chosen to participate in the experiment. Two types of observations deserve to be made here, concerning the handling of the software and the speech synthesis.

Handling. The main difficulties encountered by the pupils stemmed from their inexperience with the computer, especially editing the text: problems with placing the cursor accurately on the text, forgetting to click, etc. Nevertheless, we can see that they have rapidly familiarised themselves with theses procedures.

Speech synthesis. The excellent intelligibility of the speech synthesis results in it being at the root of few of the pupils' problems. However, when pupils comes across a word they are unfamiliar with, they do not appear able to ask the advice of the teacher, as would be the case in the classroom. This can result in misunderstandings and thus, mistakes.

5.3. Interviews

A "direct" interview is held, following the exercise. It concerns the general ergonomics of the software, speech synthesis, diction and correction.

General ergonomics. On the whole, the software is considered to be well constructed and practical to use. The pupils feel at ease.

Speech synthesis. The opinions here are mixed. They range from "good", to "an artificial voice". An exercise in order to let people get used to it could have been carried out prior to the experiment. On the other hand, research in speech synthesis is leaning towards a wider variety of voices (for example, woman's voice/man's voice, an advertising voice, or an airport voice). Maybe a teacher's voice would be a good idea.

Correction. The pupils were happy with the correction.

Preferences. One of the purposes of the interview is to evaluate the difference between Ordictée and traditional dictation. In answer to the question, "What difference do you note between this type of dictation compared to that of the teacher?" two different answers emerged: "With the teacher we can ask questions" (meaning that if we are stuck the teacher can help us). And on the other hand "We have a good control of the exercise" (meaning that we can manage our time without being dependant on the teacher).

6. Conclusion

Ordictée is a kind of software which allows pupils to practice dictation. The correction, carried out in one of three modules is based on an optimal comparison of strings using a dynamic programming algorithm. An evaluation of the pupil module has been carried out. Although at the moment only the French version is available, the technique used to carry out the correction of the dictation authorises practically immediate transport towards languages for which speech synthesis does exist. A version in Breton language is also planned.

References

D. Cotto, M. De Calmes, I. Ferrané, J.F. Malet, J.-M. Pécatte, G. Pérennou, C. Santiago. Usage didactique de produits de l'Industrie de la Langue : Un Système interactif de

dictée/correction pour l'apprentissage autonome du français. *ERGO-IA,* Biarritz, 19-21 septembre 1990, p 121-129.

D. Cotto, M. De Calmes, I. Ferrané, J.F. Malet, J.-M. Pécatte, G. Pérennou, C. Santiago. An Automatic Interactive Dictation and Correction System. Application to Language Teaching/Learning. *Proceedings of COGNITIVA 90.* AFCET, Madrid, Spain, p. 419-426. November 20-23, 1990.

M.W. Du, S.C. Chang. A model and a fast algorithm for multiple errors spelling correction. *Acta Informatica 29, 281,* p. 281-302. Springer-Verlag 1992. D. Feneuille, J.-C. Fontaine. Synthèse vocale et lecture. EA007 Cap d'Agde, ADI1987 Paris.

J.P. Fournier. Correction automatisée dans les systèmes questions- réponses en langage naturel. *CIIAM86, 1-5 décembre 1986,* Marseille. Hermès. 1986.

D. Gries, B. Burkhardt. *Presenting an Algorithm to find the Minimum Edit Distance.* TR 88-903. Cornell University. March 1988.

G. A. Stephen. *String Search.* TR-92-gas-01. School of Electronic Engineering Science. University College of North Wales. 1992.

R.A. Wagner. *On the Complexity of the Extended String-to-String Correction Problem. Time Warps String Edits, and Macromolecules: the theory and Practice of Sequence Comparison.* Addison-Wesley, 1983.

Acknowledgements. The authors would like to thank C. Delessalle, M. Gaillard, S. Mironnet, L. Egan, Primary school inspecteur at Lannion, and G. Janin, director of Saint Roch school at Lannion for their contributions to the project. This project was partly funded by ANVAR BRETAGNE.

A Conversational Agent for Food-ordering Dialog Based on VenusDictate

Hsien-Chang Wang, Jhing-Fa Wang, and Yi-Nan Liu
Institute of Informational Engineering
National Cheng Kung University
No. 1 University Road, Tainan, Taiwan R.O.C.
E-mail: {wangsj, wangjf, liuyn}@server2.iie.ncku.edu.tw

Abstract

In this paper, we introduce a conversational agent which is applied to the food-ordering dialog system. It uses VenusDictate as speech recognition front-end, then understands the semantics of the input sentence by extracting the keywords of the sentence, finally it interacts with the user by speech. The experimental results show that the performance of this agent is good in this application domain.

1. Introduction

The applications of natural language research can be divided into two major classes: text-based applications and dialog-based applications[James 1995]. Text-based applications involve the processing of written text, such as book, newspapers, Internet messages, email messages, and so on. Text-based natural language research is ongoing in applications such as extracting information from message or articles, language translation, summarizing text, etc. On the other hand, dialog-based applications involve human-machine communication which involves spoken language or interaction using keyboards. Important applications include database answering, automated customer service over the telephone, tutoring system, machine controlling, and so on [H. 1992, Ren 1991, Hsien 1997].

Dialog-based systems are quite different from text-based systems. For example, the language used is very different. Also, the system needs to interact with the user in order to maintain a natural, smooth-flowing dialog.

In this paper, a dialog-based food-ordering system is introduced. We build a conversational agent to perform necessary processes of a dialog system, including speech recognition, keyword extraction, intention and syntactic analysis, semantic understanding and proper response generation. We divided our paper into several sections. In Section 2, we brief the architecture of the dialog system. Section 3 is about the corpus collection and analysis. Section 4 introduces our speech recognition front-end -- VenusDictate and keyword extracting. Section 5 describes the analysis of intention and syntax of the input sentence. Section 6 describes how the interactive responding system operates. Section 7 is the experimental results, and we give a conclusion in Section 8.

2. System Architecture

Figure 2.1 shows the architecture of our food-ordering dialog system. The conversational agent plays an important role in our system. We implement this agent by four sub-processes -- speech recognition, keyword extraction, semantics derivation, and interactive response. The flow of a food-ordering dialog would be like this: the customer inputs the ordering sentence via the microphone, then the input speech is recognized by VenusDictate system and produces the candidate syllable lattices. These candidate syllables then passed to keyword extracting unit to acquire the keywords. Those keywords are then used to derive the semantics and thus determine the intention of the customer. Finally, the interactive response system replies proper message to the customer to complete the dialog. The detail processes of each sub-system are described in the following sections.

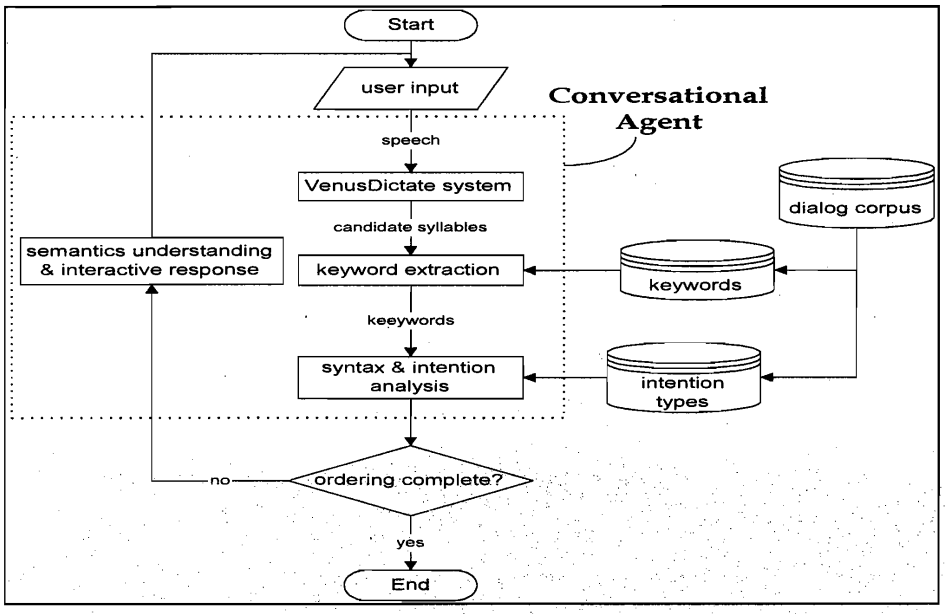


Figure 2.1 Architecture of the Food-ordering Dialog System

3. Dialog Corpus Collection and Analysis

3.1 Collecting Corpus

Dialog corpus can be divided into two major types, they are speech-format and text-format corpus. Speech-format corpora are usually used for training and evaluation of a recognition system. On the other hand, in order to analyze the syntax and semantics of the dialog, what we need is the text-format corpus.

There are two methods to collect dialog corpus. The first one is to simulate the conversation between the boss and the customer. This method has the disadvantage that the corpus will be lack of many situations in a real conversation. Also, sentences generated by such manner will probably tend to be fixed on some special patterns. The second method is to record the conversation in the food store and then transfer them into text-format corpus. We use this approach to collect corpus for our system.

Due to some reasons, we cannot persuade the boss of fast-food stores to participate our task of collecting corpus. So we choose the breakfast stores in the neighborhood of our school to record the conversation between the boss and customers. By this way, we have collected several hundreds of sentences as the corpus. The examples of these conversations can be found in the Appendix.

3.2 Keywords Classification

By analyzing the corpus, we find that some types of words play important roles in a food-ordering dialog system, such as the names of food, amount of drink, and so on. We define these words as keywords and divide them into 11 groups by their meaning. Table 3.1 lists all these 11 **keyword types** in our system. Note that we also define the abbreviation of a certain keyword as a keyword.

Keyword types	Meaning	Examples of keywords
Food	the names of food	三明治(sandwich), 漢堡(hamburger)
Drink	the names of drink	紅茶(black tea), 咖啡(coffee)
Amount	amount of food/drink	一杯(one cup), 兩個(two)
Attribute	modifier of a certain food/drink	冰的(cool), 溫的(warm)
Place	where to eat	這邊用(here), 外帶(to go)
Y_N	positive or negative modality	是的(yes), 不用(no)
Want	ordering something	我要(I want), 給我(give me)
What	ask about something	什麼是(what is),
Have	ask if there exists something	有沒有(do you have)
Price	ask for the price of something	多少(how much), 幾元(how many dollars)
NonKeyword	words without significant meaning	麻煩(would you), 哈囉(hello)

Table 3.1 Keyword types of a food-ordering dialog system

4. VenusDictate System

4.1 Introduction of VenusDictate System

VenusDictate system is a speaker dependent Mandarin word recognition system developed by Institute of Computer Science and Informational Engineering, National Cheng-Kung University [Y.W. 1991, J.S. 1991, S.H. 1990]. It allows user to input speech via microphone and then output the corresponding word candidates.

The role VenusDictate plays in our system is to transfer the input speech into the candidate syllable lattice. These syllables are then further processed to derive the semantics of the input sentence. Since VenusDictate system is now available with the Windows Application Program Interface(API) format, the integration of speech and understanding system can be easily done[J.S. 1994, H.C. 1997].

4.2 Keyword Extracting Using VenusDictate

Determining the keywords of the input sentence is an essential task in a dialog system[R.C. 1995]. In our system, the syllable lattice produced by VenusDictate is used to extract the keywords. Those keywords defined in Table 3.1 are added to the lexicon of VenusDictate, then word matching is performed by VenusDictate.

When we deal with keywords, the length of word matching is important. Consider two keywords named "咖啡(coffee)" and "咖啡奶(coffee milk)", the former is a substring of the latter keyword. If we perform keyword matching without considering the length, we will probably never be able to match the longer keyword "coffee milk", instead, the keyword "coffee" will be matched. To solve this problem, we match the longer keyword first.

When matching keywords with the syllable lattice produced by VenusDictate, the whole syllable lattice is matched first to check if there is any keyword with the same length of the input sentence. If none was matched, the length of the syllable lattice is reduced by one. Those parts of syllable lattice that are matched with keywords are removed from the syllable lattice. This process will continue until the syllable lattice becomes empty. The algorithm of this process is listed below in Algorithm 4.1.

- Input: Syllable lattice with length N.
- Output: Keywords matched.
- Method:

  Step 1. Let $k=N$, if $k=0$ then goto Step 3.

  Step 2. Try to match keyword with length k.

  if success, { $N=N-k$, output keyword, goto Step 1.}

  else, { $k=k-1$, goto Step 2.}

  Step 3. End.

Algorithm 4.1 Extracting keywords from syllable lattice

Those keywords extracted by VenusDictate are passed to the intention and syntax analysis unit for further processing.

5. Intention and Syntax Analysis

Knowing the intention of the customer is important in a dialog system. Also, the syntax is an important information for the system to decide if the input sentence is nonsense. Our system uses an approach that acquires the intention first, then checks whether the input sentence is grammatical legal.

5.1 Intention Types

After analyzing the corpus, the patterns of the ordering sentences can be divided by their meaning into five **intention types**. Those five intention types in our system are shown below.

1. ***S_What***: The sentence that contains the keyword type, ***What,*** has this intention

type. The intention of the customer is to ask about the description of something.

For example: "請問*什麼是*馬來糕？" (What is the Malay-Cake?)

2. ***S_Price***: The sentence that contains keyword type ***Price.*** The intention is to ask the price of something.

   For example: "請問紅茶*多少錢*？" (What is the price of black tea?)

3. ***S_Have***: The sentence which contains keyword type ***Have.*** The intention is to ask the existence of something.

   For example: "請問*有沒有*三明治？" (Do you have sandwiches?)

4. ***S_Want***: The sentence which contains keyword type ***Want***, or contains none of the above keyword types. The intention is to order something.

   For example: "*我要*兩個漢堡，一杯豆漿" (Give me two hamburger and one cup of soybean milk.)

5. ***Y_N***: The sentence that contains keyword type ***Y_N*** representing Yes or NO.

As described above, the five keyword types, ***What, Price, Have, Want, and Y_N***, are important when determining the intention type of an input sentence. If we can find one of these keyword types in the input sentence, we can easily determine the related intention type.

5.2 Syntax Analysis

Once the intention of the input sentence is known, we perform syntactic analysis for this sentence. The structures of ordering sentences can be described by the state transition diagrams as shown in Figure 5.1. In those diagrams, each link represents one of the keyword types in Table 3.1. The starting state is $q0$, and the ending state is q_Z. The abbreviated state transition diagram S_{order} and $S_{F/D}$ is detailed in Figure 5.2 and Figure 5.3.

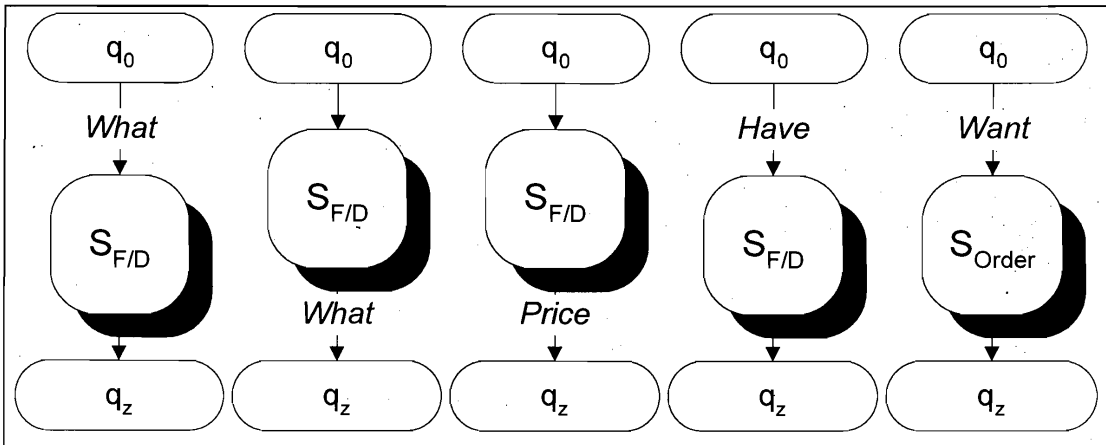


Fig. 5.1a Fig. 5.1b Fig. 5.1c Fig. 5.1d Fig. 5.1e

Figure 5.1a-5.1e Transition diagrams of intention types

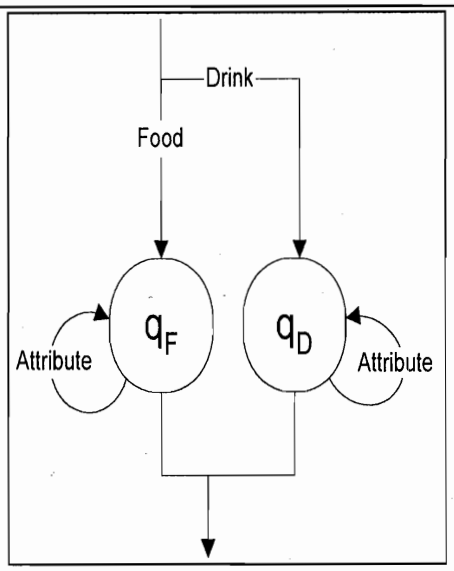


Figure 5.2

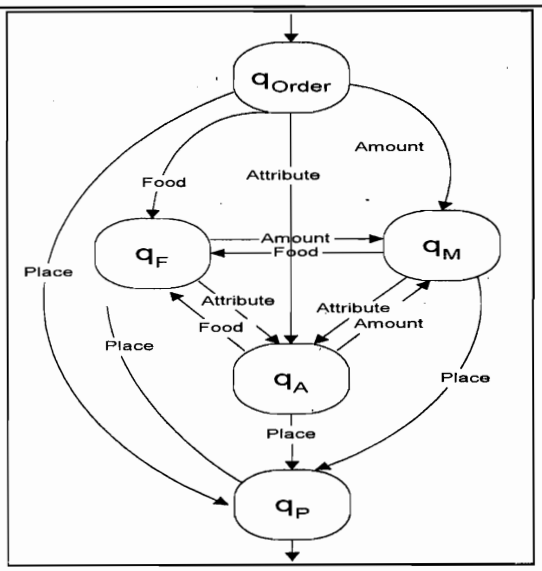


Figure 5.3

Figure 5.2 Detailed state transition diagram of $S_{F/D}$

Figure 5.3 Detailed state transition diagram of S_{order} (Note: This figure shows only ***Food*** keyword type, the figure of ***Drink*** is omitted)

We allow the keyword type ***NonKeyword*** to appear in any place of the input sentence, so we omit them in our state transition diagram to reduce the complexity of the figures.

If an input sentence can be verified by the state transition diagrams, we call it a legal sentence. For those illegal sentences, we will prompt to the user to input again. The following sentence "我要一杯是的多少錢？" (I want a cup of yes how much?") is an illegal sentence since it can not pass our state transition diagrams. For an illegal input sentence, the system asks the customer to input again.

6. Semantic Understanding and Interactive Response

The semantics of each legal input sentence contains its intention type and some extracted keyword types. For example, "Give me a cup of coffee" has the semantics:

"S_Want" + "Amount" + "Drink"

Knowing the semantics of the input sentence, we can generate proper response by considering its intentional type. The response of our dialog system is generated by combining several keywords with phrases. We will describe the response of intention types ***S_What, S_Have, S_Price,*** and ***S_Want*** respectively in the following subsections. Before that, we give some notations used in the response sentence.

- *?F/D*: The name of food or drink mentioned in the customer's ordering sentence (may include amount and attribute).

- *Description(F/D)*: The description of the mentioned food or drink.
- *Price(F/D)*: The price of the mentioned food or drink.

Note that *Description(F/D)* and *Price(F/D)* are stored in the knowledge base.

6.1 Response for Intention Type S_What

If the semantics of the input is ***S_What***, the response sentence will be generated in the form:

The ?(F/D) is Description(F/D).

For instance, the customer may ask: "What is the Malay-Cake?" Then our system will response "The Malay-Cake is *Description(Malay-Cake)*." The *Description(Malay-Cake)* is the description of Malay-Cake and is stored in the knowledge base.

6.2 Response for Intention Type S_Have and S_Price

If the input sentence has intention type ***S_Have***, the response sentence will be:

Yes, we have ?(F/D).

or

No, we do not sell ?(F/D).

If the intention type of the input sentence is ***S_Price***, the response will be:

The ?(F/D) cost Price(F/D).

6.3 Response for Intention Type S_Want

The most complex intention type of the customer is **S_Want**. When ordering, the pattern of the input sentence may vary from person to person. To handle this kind of problem, we define that if the customers want to order something, the dialog should not finish until five keyword types had been input. Those five keyword types are ***Food***, ***Drink***, ***Place***, ***Amount***, ***Attribute***.

If an input sentence lacks of some keyword types, the response system will ask the customer for those items. Consider a food-ordering dialog shown below.

Customer:	Hi, I want soybean milk.
Boss:	How many cups? Cool or Warm?
Customer:	One cup, cool.
Boss:	Do you want some food?
Customer:	No.
Boss:	Do you want to eat in the store?
Customer:	Take-out.

In the process of the interactive dialog, system determines which keyword type is absent, then generates the corresponding query sentence to ask the user to input that keyword type. The dialog will not finish until all five keyword types are all filled. Table 6.1 shows how this is done.

	Food	Drink	Attribute	Amount	Place
customer		soybean milk			
system			ask Attribute	ask Amount	
customer			cool	one cup	
system	ask Food				
customer	No				
system					ask Place
customer					take-away

Table 6.1 Illustration of how interactive response system work.

When the ordering dialog is complete, our system repeats the customer's order and totals the price of food and drink. To make the system more friendly, the response sentences are chosen from some predefined sentences randomly. For instance, in the beginning of the dialog, the greeting of the boss may be "Welcome", "Hello, what do you want", or "What can I do for you?" etc. Furthermore, if the customer has just ordered food, the response sentence may be "Do you want some drink?" or "Take-out?". In this way, we make our response sentence more flexible.

7. Experimental Results

We implement our food-ordering dialog system by Microsoft Visual C++ 4.0, and integrated it with the VenusDictate system. The platform is a Pentium 120 PC with Windows 95 operating system.

Firstly, we test the performance of the single keyword recognition rate of the VenusDictate system. The tester pronounces each keyword of 11 keyword types three times. Then we calculate the Top 1 correct rate. VenusDictate system allows the user to input speech by two pronunciation manners, word-connected and semi-continuous. Our tests include both input methods. The Correct-Rate-I is the recognition rate of word-connected input method, and Correct-Rate-II is that of semi-continuous method. The result is shown in Table 8.1. The recognition rate of the word-connected version is better than that of the semi-continuous version of VenusDictate. The reason is that continuous speech recognition sometimes causes insertion or deletion problems.

	Food	Drink	Amount	Attribute	Place	Y_N	What	Have	Price	Want	Non-keyword	Average correct rate
Correct-Rate-I	95.5	99.7	89.5	83.3	96.4	88.7	93.7	95.7	98.5	92.6	94.6	93.4
Correct-Rate-II	80.5	73.1	84.4	61.1	87.7	73.5	86.7	81.7	88.5	90.2	82.5	80.9

Table 7.1 Single keyword Recognition rate of VenusDictate

Secondly, to test the performance of our system, we randomly choose 50 food ordering dialogs from the corpus to be tested. Since our system allows the user to complete his order in one sentence (***fully***) or in several sentences (***partially***), our test contains these two types of input methods. The result of "fully" and "partially" input method is shown in Table 8.2.

There are two kinds of test performed for both "fully" and "partially" input methods. The first one is the number of success within one trial which is abbreviated as SW1T. It means that the tester successfully orders his food in the first trail via VenusDictate. The second one is SW3T(success within 3 trials) which means the order succeeds within three trials via VenusDictate. Note that the SW3T includes the SW1T ones. From Table 8.2, we find that the correct rate of fully ordering method is poor than that of partially one. The reason is that a fully order contains longer input speech, and may cause more recognition errors.

	testing sentences	SW1T		SW3T	
		# of correct sentences	correct rate	# of correct sentences	correct rate
fully	50	25	50%	40	80%
partially	50	36	72%	46	92%

Table 7.2 Experimental result of our system.

8. Conclusion

In this paper, we describe the implementation of a conversational agent for food-ordering dialog system. We use VenusDictate as the speech recognition front-end, then determine the syntax and intention of the input sentence, finally generate proper response to interact with the customer.

The conversational agent proposed in this paper has a flexible architecture. The speech recognition front-end can be replaced by another speech recognition system, such as a speaker independent recognition system or a recognition system which works over the telephone-network. Also, a text-to-speech system can be easily integrated into this agent.

The collection of dialog corpus is a difficult task which costs much money and manpower. However, in order to establish a practical dialog system, it is an unavoidable important task. We wish that there will be more manpower invested into this task and the collected corpus can be shared.

There are many dialog-based applications, such as automated service over the telephone, tutoring system, etc. With the experience of building this food-ordering system, we hope to develop an automatic or semi-automatic system which can help to transplant from one application domain to another easily.

References

James Allen, Natural Language Understanding, The Benjamin/Cummings Publish Company, INC. 1995

H. Tsuboi and Y. Takebayashi, "A real-time task-oriented speech understanding system using keyword spotting," Proc. ICASSP, pp.197-200,1992.

Ren-Jong Hseu, "Automatic Chinese Telephone Operator Assistant, ACTOA," Proceedings of ROCLING IV, pp.167-191, 1991.

Hsien-C. Wang, Jhing-F. Wang and Din-Y. Liou, "Natural Language Understanding for Telephone Transfer Dialogue", Proceeding of ICCPOL '97, pp. 7-12.

Y.W. Jeng, J. F. Wang, "Large Vocabulary Size Speech Recognition System", master thesis, Inst. of Info. Eng., Natl. Cheng Kong Univ. 1991.

J. S. Shyuu, J. F. Wang, "A Speaker Independent Continuous Mandarin Digit Recognition System", master thesis, Inst. of Info. Eng., Natl. Cheng Kong Univ. 1991.

S. H. Lee and H. J. Lee, "A Unification-Based Approach for Chinese Inquiry Sentences Processing," Proceedings of ROCLING III, pp.441-466, 1990.

J. S. Shyuu, J. F. Wang and C. H. Wu, "An user friendly interface, high reliability, large vocabulary Mandarin speech recognition system", National human interface speech communication conference III, China, 1994.

H. C. Wang, J. S. Shyuu, and J. F. Wang, "Natural Language Understanding Based on VenusDictate", Proceeding of CSIA '97, pp. 185-190.

R. C. Rose, "Keyword detection in conversational speech utterances using hidden Markov model based continuous speech recognition," Computer Speech and Language 9, 309-333, 1995.

Truncation on Combined Word-Based and Class-Based Language Model Using Kullback-Leibler Distance Criterion

Kae-Cherng Yang[1], Tai-Hsuan Ho[2], Juei-Sung Lin[2], Lin-Shan Lee[123]

[1]Department of Electrical Engineering, Nation Taiwan University, Taipei, R.O.C.
[2]Department Computer Science and Information Engineering, Nation Taiwan University, Taipei, R.O.C.
[3]Institute of Information Science , Academia Sinica, Taipei, R.O.C.
E-Mail: bangdoll@speech.ee.ntu.edu.tw Tel: 886-2-369-2535

Abstract

In this paper we present a novel approach to truncate combined word-based and class-based n-gram language model using Kullback-Leibler distance criterion. First, we investigate a reliable backoff scheme for unseen n-gram using class-based language model, which outperforms conventional approaches using (n-1)-gram in perplexity for both training and testing data. As for the language model truncation, our approach uses dynamic thresholds for different words or word contexts determined by the Kullback-Leibler distance criterion, as opposed to the conventional scheme which truncates the language model by a constant threshold. In our experiments, 80% of the parameters are reduced by using the combined word-based and class-based n-gram language model and the Kullback-Leibler distance truncation criterion, while the perplexity only increases 1.6%, as compared with the word bigram language model without any truncation.

1. Introduction

In the large vocabulary continuous speech recognition, the n-gram language model has been widely used as the effective linguistic constraint to determine the final transcription among several text hypotheses. In order to get a reliable language model, we need a lot of text data and therefore the size of a language model will be also very large. However, due to the constraint of memory, a huge language model will make the speech recognition system impractical. Thus reducing the language model size is important.

An intuitive approach to reduce the language model size is to truncate k-gram entries that appear below a given threshold in the training corpus. Another common approach use the class-based n-gram language model (Brown 1992, Jardino 1993, Martin 1993), which is

intrinsically more compact and outperforming the word-based model at estimating unseen word sequences. However, given enough training data, the performance of the word-based model usually surpasses that of the class-based model because it is more accurate in capturing sequential relationships between particular words.

To keep the advantage of word-based and class-based language models, combining these two models within the backoff probability estimation phase is a good approach (Niesler 1996). By using the class-based model as the backoff estimation instead of the lower order word-based model, the performance is apparently improved. Furthermore, with this more accurate backoff estimation using class-based model, we can impose a heavier truncation on word-based model, which only slightly degrades the performance. Therefore, a combined model of both word-based and class-based model for backoff estimation under heavy truncation could meet the high accuracy and compact memory storage requirement at the same time, and this is our approach. Another advantage of this combined model we proposed is that its performance is always higher than using class-based language model alone. Even in the worst case, it still performs as well as the class-based model. That is, if all word n-gram entries have been truncated, this combined model will be the same as the class-based model alone. From this viewpoint, heavy truncation can be done since the lowest bound of performance can also keep in the level of class-based models.

In order to get a better truncation for a given amount of parameters, we use the Kullback-Leibler distance criterion (Kneser 1996, Kullback 1958) to determine the thresholds for all k-gram entries where $k < n$. In our truncating procedure, if $N(w_i,w_{i+1},\ldots, w_{i+k}) < Th(w_i,w_{i+1},\ldots,w_{i+k-1})$, then the context entry $(w_i, w_{i+1}, \ldots, w_{i+k})$ will be deleted, where $N(w_i,w_{i+1},\ldots,w_{i+k})$ is the occurrence count of word context $(w_i, w_{i+1}, \ldots, w_{i+k})$ in training corpus, and $Th(w_i,w_{i+1},\ldots,w_{i+k-1})$ is the threshold given context $(w_i,w_{i+1},\ldots,w_{i+k-1})$.

The rest of this paper is organized as follows: Section 2 describes the language model which combining word-based and class-based models; Section 3 describes the truncating criterion named as Kullback-Leibler distance; Section 4 describes the algorithm of truncation using Kullback-Leibler distance criterion; Section 5 presents the experimental results of the perplexity measures. Finally, in Section 6 we will give a brief conclusion.

2. Combined Language Model

This language model combines word-based and class-based models within the backoff framework. The conventional n-gram probability estimated by maximum-likelihood approach has been proven very effective for modeling language. However, word sequence not present in the training corpus will result in zero probability for the test data. Therefore, we need backoff scheme to calculate the probability for unseen events. Briefly, when we compute the likelihood of word contexts, a certain amount of the total probability mass for the conditioning context should be redistributed to the unobserved words. In the conventional model, the redistribution is proportional to the probability from the next lower-order model. However, from past experiences, we know that the class-based language model is more robust for estimating the probabilities for unseen events. Based on this concept, we believe that using class-based language model in backoff phase can make more accurate estimation among unseen word sequences.

In this combined model, the probability estimation formula of a given word context with n words $w_{i-n+1}, \ldots, w_i$ is as follows:

$$P_n(w_i \mid h = w_{i-1},\ldots,w_{i-n+1}) = \begin{cases} P_{w,n}(w_i \mid h) & \text{if } w_i \in W_n(h) \\ \alpha(h) P_{c,n}(w_i \mid C(h)) & \text{otherwise} \end{cases} \quad (1)$$

where

- $h = w_{i-n+1}, \ldots, w_{i-1}$ means the word history. For example of trigram model, $h = w_{i-2}, w_{i-1}$.
- $W_n(h)$ is the set of words which connect to word context h in training corpus, i.e., if word $w \in W_n(h)$, it means that there is an n-gram entry that stores the word context (h,w) and its count.
- $P_{w,n}(w \mid h)$ is the word conditional probability given h for which w belongs to $W_n(h)$, i.e., the n-gram entry (h,w) exists in the word-base model. The estimation of $P_{w,n}(w \mid h)$ will use both word-based and class-based models.
- $\alpha(h)$ is the backoff weight for the given history h. In our combined language model, linear backoff (Placeway 1993) was employed.
- $P_{c,n}(w \mid C(h))$ is the word conditional probability given $C(h)$ where $C(h)$ is the class sequence of words in the history. The estimation of $P_{c,n}(w \mid C(h))$ uses class-based

language model only.

The linear backoff approach (Placeway 1993) is a robust and simple method that can be regarded as a HMM grammar structure (Fig. 1). In contrast to conventional backoff scheme (Katz 1987), this approach estimates the probability by a linear combination of direct estimating path and backoff path.

To see the formula of the linear backoff approach, firstly, we define two terms $P_{w|h}(w \mid h)$, the direct estimation probability, and $\alpha(h)$, the backoff probability mass, as follows:

$$P_{w|h}(w|h) = \frac{N(h,w)}{N(h)+R(h)} \qquad (2)$$

$$\alpha(h) = \frac{R(h)+T(h)}{N(h)+R(h)} \qquad (3)$$

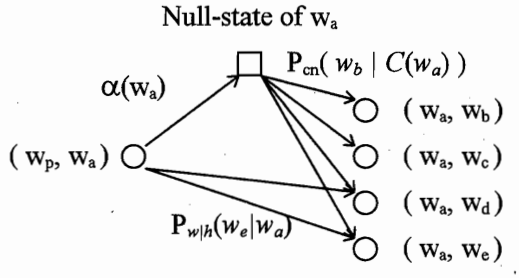


Fig. 1: Example of HMM Bigram Grammar Structure

where N(h,w) is the number of times that word w occurred behind context h in training corpus, R(h) is the number of distinct words that occurred behind context h, and T(h) is the count of total truncated entries, i.e., T(h) = $\Sigma_{w'}$ N(h,w') where w' is the word that the entry (h,w') has been truncated. Note that we should not modify N(h) and R(h) after truncating, i.e., the values of N(h) and R(h) are conditional on whole training corpus and they are independent of truncating process.

In the equations above, $P_{w|h}(w \mid h)$ is the direct estimation probability of word w given the history h and $\alpha(h)$ is the total probability mass through backoff path (null-state) in Fig. 1. For the unseen events, e.g., (w_a, w_b) and (w_a, w_c) in Fig. 1, the probabilities are all estimated by going through the backoff path. As for the observed events, e.g., (w_a, w_d) and (w_a, w_e) in Fig. 1, the probabilities are estimated not only through a direct arc path but also a null-state path. Thus the equation of $P_{w,n}(w \mid h)$ is as follows:

$$P_{w,n}(w|h) = P_{w|h}(w|h) + \alpha(h) P_{c,n}(w|C(h)) \qquad (4)$$

where C(h) is {C(w_{i-n+1}), ..., C(w_{i-1})} and C(w) is the class of word w. We assign $P_{c,n}(w \mid C(h))$ as follows:

$$P_{c,n}(w|C(h)) = P(w|C(w)) P_n(C(w)|C(h)) \qquad (5)$$

where

$$P(w|C(w)) = \frac{N(w)}{N(C(w))} \tag{6}$$

$$P_n(C(w)|C(h)) = P_{c(w)|c(h)}(C(w)|C(h)) + \alpha(C(h))P_{n-1}(C(w)|C(h-1)) \tag{7}$$

The estimation of $P_{C(w)|C(h)}(C(w)|C(h))$ and $\alpha(C(h))$ is the same as that for $P_{w|h}(w|h)$ and $\alpha(h)$ except that the word and word history sequence number become class and class sequence history number in training corpus and the backoff weight $\alpha(C(h))$ is estimated by lower order class-based language model. Although the above model is complex, we can prove that the summation of probabilities equal to one for all words given the history.

3. Truncation Using Kullback-Leibler Distance Criterion

In order to have a better truncation result, we exploit the Kullback-Leibler distance criterion to measure the quality of truncated language model and determine thresholds for all word k-gram entries where k < n. Let P_I denote the probability distribution of initial language model without any truncation and P_T denote the probability distribution of truncated language model. The Kullback-Leibler distance of these two models is as follows:

$$D(P_T;P_I) = \sum_{h,w} P_I(h,w)\log\frac{P_I(w|h)}{P_T(w|h)} \tag{8}$$

where $P_I(w|h) = P_n(w|h)$ in the initial model and $P_T(w|h) = P_n(w|h)$ in the truncated model.

We can show that $D(P_T;P_I)$ will be greater than or equal to zero and the equality holds if $P_I(w|h) = P_T(w|h)$ only. There is one assumption for using Kullback-Leibler distance criterion: the initial language model without truncation will be the best model comparing to truncated models. Thus if we do any truncation, the resulted model will be worse than initial model. Under this assumption, if the distance of a truncated model is lower, the concerned model with P_T is more near to initial model P_I and therefore the model is considered to be better.

For each time we truncate the k-gram entries, if we use equation (8) to compute the distance of initial model and truncated model, the computation cost is much expensive. To

reduce the computational complexity, we can further derive equation (8). Let h_k denote the first k word context in history h. The equation (8) can be rewritten as follows:

$$D(P_T;P_I)=\sum_{(h,w),h_k\neq h_k'}P_I(h,w)\log\frac{P_I(w|h)}{P_T(w|h)}+\sum_{(h,w),h_k=h_k'}P_I(h,w)\log\frac{P_I(w|h)}{P_T(w|h)} \quad (9)$$

If we change the threshold of k-gram entry with history h_k', we can only calculate the later term in equation (8). Thus we can define the term d(h_k) as follows:

$$\begin{aligned} d(h_k') &= \sum_{(h,w),h_k=h_k'}P_I(h,w)\log\frac{P_I(w|h)}{P_T(w|h)} \\ &= \sum_{(h,w),h_k=h_k'}P_I(h,w)\log P_I(w|h)-\sum_{(h,w),h_k=h_k'}P_I(h,w)\log P_T(w|h) \quad (10) \\ &= d_0(h_k')-d_T(h_k') \end{aligned}$$

We can calculate the former term in equation (10), d_0 (h_k'), in the initialization procedure and store them. For each time we adjust the threshold, the later term $d_T(h_k')$ is the one that we must calculate .

4. Truncating Algorithm

In the practical system, there is a constraint of the memory that we can use. Therefore the total parameter number of all k-gram entries has an upper bound. Our algorithm is to find the better solution to determine what parameters in word-based language model should be truncated. The parameters in class-based model will not be truncated because the class-based language is much smaller than word-based language model and they are robust to calculate the backoff probabilities.

Before describing our algorithm, firstly we define some terminology as follows:

- Th(h_k'): the threshold of k-gram entries with word context h_k'. If one (k+1)-gram entry with history h_k' is that its count occurring in training corpus is less than Th(h_k'), this entry will be deleted and its count will be added to T(h_k') as describing in backoff phase.

- N_T(h_k' , Th): the total number of m-gram (m = k+1 ~ n) entries with first k symbol history equaling to h_k' and their counts in training corpus are less than Th.

- N_T: the total number of entries that has been deleted, i.e., $N_T = \Sigma_{h_k'} N_T(h_k', Th(h_k'))$.
- N_I: the total number of entries in the initial language model.
- $dn_T(h_k', Th) = d_T(h_k') / (N_T(h_k', Th) - N_T(h_k', Th-1))$: the normalizing distance of h_k' given threshold Th.

The algorithm is as follows:

1) Initialize
   1.1) Set all thresholds, $Th(h_k')$, equal to 2 and total truncated entry number N_T=0.
   1.2) Calculate $d_0(h_k')$, $d_T(h_k')$, and normalizing distance $dn_T(h_k', Th(h_k'))$ for each h_k'.
2) Loop
   2.1) Find the best h_k' that has the smallest distance $dn_T(h_k', Th(h_k'))$ and let $h_B = h_k'$.
   2.2) Calculate N_T. If $N_I - N_T$ <= upper bound of parameter number, then break.
   2.3) Set $Th(h_B) = Th(h_B) + 1$. Calculate N_T and $dn_T(h_B, Th(h_B))$.
3) End

Instead of the stop condition for the loop in above algorithm, we can also change the stop condition to control the performance of our combined language model. For this case, we don't need the term N_T, but we must have one term ΔE that is the accumulative distance of truncated model and initial model. If ΔE is larger than a threshold max-ΔE, then we stop the loop.

5. Experimental Results

5.1 Experimental environment

The corpus in our experiments is obtained from newspapers of eight months. Seven out of eight months' data is used for training, and the remaining one is used for testing. The lexicon is provided by CKIP. The vocabulary size is 94188 and the maximum length of word entries is nine. After the word segmentation, there are 10,136,783 words in the training corpus and 1,521,867 words in the testing corpus. The resulted word bigram language model has 2,484,757 bigram entries and 55,380 unigram entries. The perplexity of this model is 307.641. All following experiments use testing corpus to evaluate perplexities.

5.2 Class-Based Bigram Language Model

Our class-based language model is generated by two phases. In the first phase, we use simulated annealing approach [2] to cluster words. However, the result of the first phase is not good enough. In the second phase, we use the clustering result of first phase to be the initial condition and use k-means-style algorithm [3] to improve it.

In both algorithms, we classify 27,829 highest frequency words for three class models with 999, 499, and 249 classes respective, and the words in the residual part are all collected into one class. Thus the total numbers of classes are 1000, 500, and 250.

For the simulating annealing algorithm, the parameters (T_0, T_f, α, i_{max}, r_{max}) are set to (1, 10^{-100}, 0.9, 20000,5000) empirically. It takes at most 48.0 CPU hours on a Pentium 166 machine with 128M ram. For the k-means-style approach, the time complexity is larger. For the case of 1000 classes, we need 5 days for 10 iterations.

Table 1 show perplexity values for simulating annealing approach and k-means-style approach in second phase. The results show that the performance was improved after second phase process.

class number	*250*	*500*	*1000*
Simulating Annealing	538.326	469.951	412.632
k-means-style algorithm	513.094	450.246	392.211

Table 1. Perplexity measures of class-based bigram language model

5.3 Combined Bigram Language Model

The combined bigram language models discussed in section 2 are generated by combining word-based bigram model and class-based bigram models that have 250, 500, 1000 classes respectively. The perplexity values with no truncation are shown in the table 2.

Class number	*250*	*500*	*1000*	*Word-Bigram*
Parameter number	60,030	216,270	620,087	2,540,137
Perplexity value	291.724	292.260	294.523	307.641

Table 2. Perplexity measure of combined bigram and word bigram language model

Note that combined bigram language model with 1000 classed is not the best. Inversely, the combined language model with 250 classes is better than the other two models. Since our

language model structure is as HMM that contains a null-state path for not only unseen events but also observed events, there will be an overestimation problem if class number is too large. Therefore it still exists a problem to choose the best class number.

5.4 Truncation on Word Bigram Models with Constant Threshold

We truncate the bigram entries that their counts are smaller than a given threshold and then calculate the perplexity values for word bigram model. The experimental results are shown in table 3.

	Word bigram model	
threshold	Total number of parameters	perplexity value
2	541,014	346.461
3	387,483	363.109
4	300,430	378.260
5	245,734	391.455
6	207,909	403.767
8	158,931	424.919
10	128,077	444.307
20	63,753	515.61

Table 3. Parameter number and perplexity value for word bigram models with constant threshold

5.5 Truncation on Word and Combined Language Models Using Kullback-Leibler Distance

We truncate the bigram entries on both word-based and combined language models by Kullback-Leibler distance criterion. The entry numbers for both models are near the result of constant threshold. The parameter numbers and perplexity values are shown in table 4.

Word bigram model		combined bigram model	
Total number of parameters	perplexity value	Total number of parameters	perplexity value
541,011	347.339	540,992	315.706
387,468	362.261	387,480	326.210
300,425	375.658	300,424	335.350
245,731	387.655	245,733	344.157
207,909	398.966	207,909	352.622
158,931	417.654	158,930	367.882
128,076	434.089	128,077	383.165
63,753	496.737	63,752	485.330

Table 4. Parameter number and perplexity value for word and combined bigram models with dynamic threshold determined by Kssullback-Leibler distance where class number is 250.

Comparing table 4 with table 3, using Kullback-Leibler distance criterion to determine thresholds for all words will be better than constant threshold under the same number of parameters, especially when the total number of truncated entries is large. Besides, the combined language model is better than other two models. The perplexity of combined model is about less than 10% of word model.

6. Conclusion

In this paper, we have present the combined word-based and class-based language model within backoff framework. Our experiments show that this combined language model is better than conventional n-gram language models. Besides, for a practical system, the number of parameters in a language model can not be too much. We develop a truncation algorithm based on Kullback-Leibler distance criterion that show that the resulted model will outperform the model truncated by constant threshold. Finally, in order to get a good trade-off between complexity and performance, we show that the truncation on combined word-based and class-based n-gram language model using Kullback-Leibler distance criterion will have better results.

References

P.F. Brown, V.J. Della Pietra, P.V. deSouza, J.C. Lai, R.L. Mercer: "Class-Based n-gram Models of Natural Language", Computational Linguistics, Vol. 18, No. 4, pp. 467-479, 1992

M. Jardino, G. Adda, "Automatic Word Classification Using Simulated Annealing", Proc. ICASSP II, 1993, Minneapolis, Minnesota, USA, pp. 41-44

S. Martin, J. Liermann, H. Ney, "Algorithms for Bigram and Trigram Word Clustering", EUROSPEECH, 1995, Madrid, September, pp. 1253-1256

T.R. Niesler, P.C. Woodland, "Combination of Word-Based and Category-Based Language Models", ICSLP, 1996, vol I, pp. 220-223

R. Kneser, "Statistical Language Modeling Using a Variable Context Length", Proc. ICSLP, 1996, vol I, pp494-497

S. Kullback, *Information Theory and Statistics*, New York, NY: Wiely, 1958

P. Placeway, R. Schwartz, P. Fung, L. Nguyen, " The Estimation of Powerful Language Models from Small and Large Corpora", Proc. ICASSP, 1993, vol. II, pp. 33-36

S.M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", IEEE Trans. Acoustic, Speech, Signal Processing, vol. ASSP-35, no. 3, pp.400-401, Mar. 1987

Recognizing Korean Unknown Proper Nouns by Using Automatically Extracted Lexical Clues

Bong-Rae Park, Young-Sook Hwang, Hae-Chang Rim
Natural Language Processing Lab,
Department of Computer Science and Engineering, Korea University,
Anam-Dong, Seoul 136, Republic of South Korea
pbr@nlp.korea.ac.kr, yshwang@nlp.korea.ac.kr, rim@nlp.korea.ac.kr

Apstract

This paper presents a method of extracting lexical clues automatically from a very large corpus and recognizing unknown proper nouns by using those lexical clues. This method collects proper noun candidates from the raw corpus and extracts the lexical clues among the adjacent known words of the proper noun candidates. And then, it recognizes unknown nouns and determines whether the identified unknown noun is a proper noun or not by using its adjacent lexical clues. Experimental result shows that the proposed method can extract 1,416 lexical clues from about ten million word size corpus and can recognize unknown proper nouns in the test corpus in 92% precision rate and 72% recall rate respectively.

1. Introduction

Many current application systems of natural language processing have been developed based on the assumption that all words within texts are registered in a machine-readable dictionary. But, this assumption is wrong because there are many unknown words in real texts (Park 1997) (Lee 1995) (Weischedel 1993).

In Korean, an unknown word can be a proper noun, an affix-derived word, or a foreign noun, etc. Each kind of unknown words has different problems in being recognized. Especially, recognition of an unknown proper noun has two critical problems. The first problem is that an unknown proper noun is difficult to recognize in a word level analysis because a proper noun is classified according to its meaning rather than its grammatical function. Moreover, the Korean proper nouns don't have any surface features unlike the other language; English proper nouns use an uppercase initial which is useful to recognize unknown proper nouns, but Korean proper nouns don't have such a property. Therefore, recognition of unknown proper nouns requires a kind of context analysis beyond a word level analysis. And the second problem is that many proper nouns temporarily appear on texts, so it is inappropriate to register

them in a dictionary even if they are recognized. If we register temporarily used personal names or place names in a lexical dictionary as soon as they are recognized, then the dictionary becomes inefficiently large and causes many improper morphological analyses (Park 1995) (Atwell 1987). Accordingly, unknown proper nouns must be recognized in a real time without depending on a dictionary.

1.1. Existing Approach

Two existing methods are well known for dealing with Korean unknown proper nouns. The first method is to split a josa[1] from an eojeol[2] which fails to be morphologically analyzed and then to regard the head of the eojeol as an unknown proper noun. And the second method tries to recognize unknown proper nouns by using manually extracted lexical clues.

The first method is based on the assumption that all eojeols including unknown proper nouns fail to be morphologically analyzed and all eojeols which fail to be mophologically analyzed include unknown proper nouns. However, we observed that some eojeols which fail to be morphologically analyzed do not include unknown proper nouns but the other kinds of unknown words or orthographic errors. Moreover, about 10% of eojeols including unknown proper nouns can be improperly analyzed[3], and the last syllable of an unknown proper noun can often be mistaken for the first syllable of a josa (Park 1997). Therefore, this method suffers from some difficulties in recognizing unknown proper nouns.

And the second method recognizes an unknown proper noun by using their adjacent lexical clues. In this method, lexical clues are manually extracted by human experts in advance. Therefore, this method requires labor intensive work (Yang 1996). Recently, a semi-automatic method has been reported to extract more lexical clues by using some reliable lexical clues which are prepared by human experts (Strzalkowski 1996). In this paper, we present an automatic method of extracting lexical clues.

1.2. System Overview

Our method of recognizing unknown proper nouns consists of two stages as shown in Figure 1. The first stage is to extract lexical clues, and the second stage is to recognize unknown proper nouns by using those lexical clues.

---

[1] A josa is a tail combined with a noun head in Korean.

[2] An eojeol is a spacing unit in Korean like a word in English. An eojeol consists of one or more morphemes. It sometimes corresponds to a word or a phrase in English.

[3] In our test corpus, 9.8% of eojeols including unknown proper nouns are improperly analyzed.

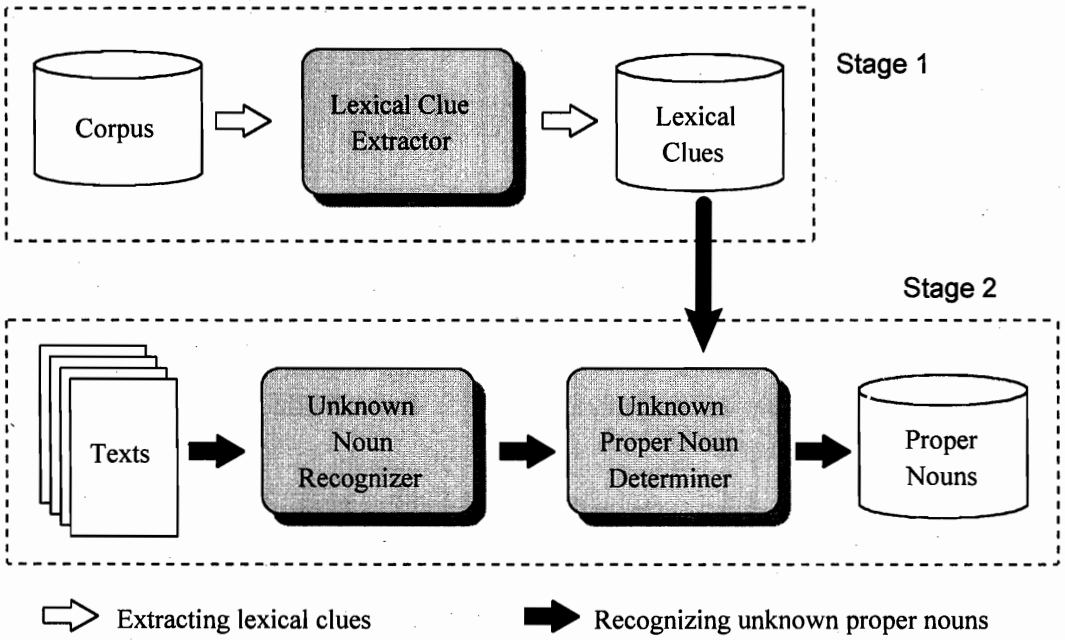


Figure 1. System configuration

The first stage is implemented by lexical clue extractor. The lexical clue extractor collects all eojeols which fail to be morphologically analyzed, and selects eojeols which include proper noun candidates. And then it extracts lexical clue candidates from the adjacent known words of the proper noun candidates, and determines whether each lexical clue candidate is a real lexical clue by estimating its degree of coupling with proper nouns. The degree of coupling with proper nouns is estimated by using two probabilities: the probability that the lexical clue candidate occurs in a set of the eojeols including the proper noun candidates and the probability that the lexical clue candidate occurs in an entire corpus. If the former probability is larger than the latter probability, the lexical clue candidate is determined to be a real lexical clue which can be used to recognize unknown proper nouns.

And the second stage is implemented by two processors: unknown noun recognizer and unknown proper noun determiner. The unknown noun recognizer is a kind of a preprocessor for recognizing unknown proper nouns and its function is to extract unknown words from unknown eojeols and identify unknown nouns among the extracted unknown words. And then, the proper noun determiner determines unknown proper nouns among the unknown nouns. In other words, this processor detects unknown nouns which occur together with one or more lexical clues and determines whether each unknown noun is an unknown proper noun by using its degree of coupling with the adjacent lexical clues.

2. Extracting Lexical Clues

Our method recognizes unknown proper nouns by using their lexical clues. Therefore, the extraction of qualified lexical clues has a good effect on precision and recall rates of recognizing unknown proper nouns. This section presents a method of extracting lexical clues automatically from a very large raw corpus. First, we collect eojeols which are expected to include unknown proper nouns. Most eojeols which include unknown proper nouns fail to be morphologically analyzed. So, we gather all eojeols which fail to be morphologically analyzed, and then filter out the eojeols including affix-derived words[4] or spacing errors (Park 1995) from them[5]. The eojeols including affix-derived words can be found by the analysis of one-syllable affix[6]. Also, the eojeols including spacing errors can be detected by using the existing method of detecting spacing errors. Accordingly, the remaining eojeols are expected to include unknown proper nouns.

In Korean, an eojeol can be splitted into a head and a tail, in which a head consists of one or more lexical morphemes and a tail consists of zero or more grammatical morphemes. A proper noun and a lexical clue are often combined to become a head because they are lexical morphemes. Therefore, we split heads from the above collected eojeols which are expected to include proper nouns and select only heads having both a proper noun and a lexical clue.

Figure 2 shows a detailed algorithm of collecting those heads and extracting lexical clues from those heads. In the steps 1 to 3, we collect eojeols which fail to be morphologically analyzed and filter out eojeols including affix-derived words and spacing errors from them, and so remaining eojeols are expected to include only proper nouns. And then, in the steps 4 to 8, we extract lexical clue candidates from the adjacent known words of the proper noun candidates and measure their occurrence probabilities that the lexical clue candidates occur near proper noun candidates. And in the steps 9 to 11, we measure the occurrence probabilities that the lexical clue candidates occur in the entire corpus. Finally, in the step 12, we compare two occurrence probabilities of each lexical clue candidate and extract the lexical clues which occur more often near proper noun candidates than the other words.

---

[4] In Korean, affixes are very diverse and difficult to distinguish from the other words.

[5] In Korean, most eojeols which fail to be morphologically analyzed include an unknown proper noun, an unknown affix-derived word, or a spacing error.

[6] Generally, one-syllable affix analysis has not been performed in Korean language processing systems because it causes the overgeneration problem.

// Collecting eojeols including proper nouns

1: Collect all eojeol **E**'s that fail to be morphologically analyzed from corpus **R**.

2: Exclude eojeol **E**'s with one syllable affix.

3: Exclude eojeol **E**'s having any spacing error.

// Extracting lexical clue candidates and their occurrence probabilities

4: Extract head **H**'s from the remaining eojeol **E**'s.

5: Exclude head **H**'s whose length is less than four syllables.

6: Construct a set X with head **H**'s.

7: Extract lexical clue candidate **c**'s from each head **H** in the set **X**.

8: Estimate a probability **P(c|X)** for each lexical clue candidate.

// Estimating occurrence prob.'s of the lexical clue candidates in a raw corpus

9: Extract all head **S**'s from all eojeols within the corpus **R**.

10: Construct a set **A** with heads **S**'s.

11: Estimate a probability **P(c|A)** for each lexical clue candidate **c**.

// Selecting lexical clues by comparing above two kinds of probabilities.

12: Select lexical clue candidates according to the following formula.

$$\frac{P(c|X)}{P(c|A)} > 1$$

Figure 2. An algorithm of selecting lexical clues

3. A method of recognizing unknown nouns

In Korean, unknown words can be classified into nouns, verbs and adverbs according to their part-of-speeches. The number of unknown verbs and adverbs are small, but they can not be neglected, and a proper noun is a kind of a noun. Therefore, unknown nouns must be recognized from the unknown words before we try to recognize unknown proper nouns.

The existing methods of recognizing unknown nouns detect eojeols which fail to be morphologically analyzed and generate every possible unknown word candidates from the eojeols and then select optimal unknown word candidates by using stochastic and/or linguistic informations. However, the existing methods have three critical problems. The first problem is that those methods can't detect any unknown word candidates from improperly analyzed eojeols which include unknown words, and the

second problem is that those methods often overgenerate unknown word candidates from eojeols which fail to be morphologically analyzed. And third problem is that those methods have difficulty in splitting an unknown noun and known words in the same eojeol.

Figure 3 shows the basic idea of the existing methods[7]. In this figure, eojeols 이순신의[*lee-sun-sin-eui*] and 원정가면[*won-jeong-ga-myeon*] have unknown words 이순신[*lee-sun-sin*] and 원정가[*won-jeong-ga*] respectively, but those unknown words are not detected because the eojeols are improperly analyzed[8]. And two or more unknown word candidates are overgenerated from eojeols 원정가서도[*won-jeong-ga-seo-do*] and 이순신장군만[*lee-sun-sin-jang-gun-man*] which fail to be morphologically analyzed. Moreover, the unknown noun 이순신[*lee-sun-sin*] is not exactly extracted from the eojeol 이순신장군만[{*lee-sun-sin-jang-gun-man*], but 이순신장군[*lee-sun-sin-jang-gun*] is extracted.

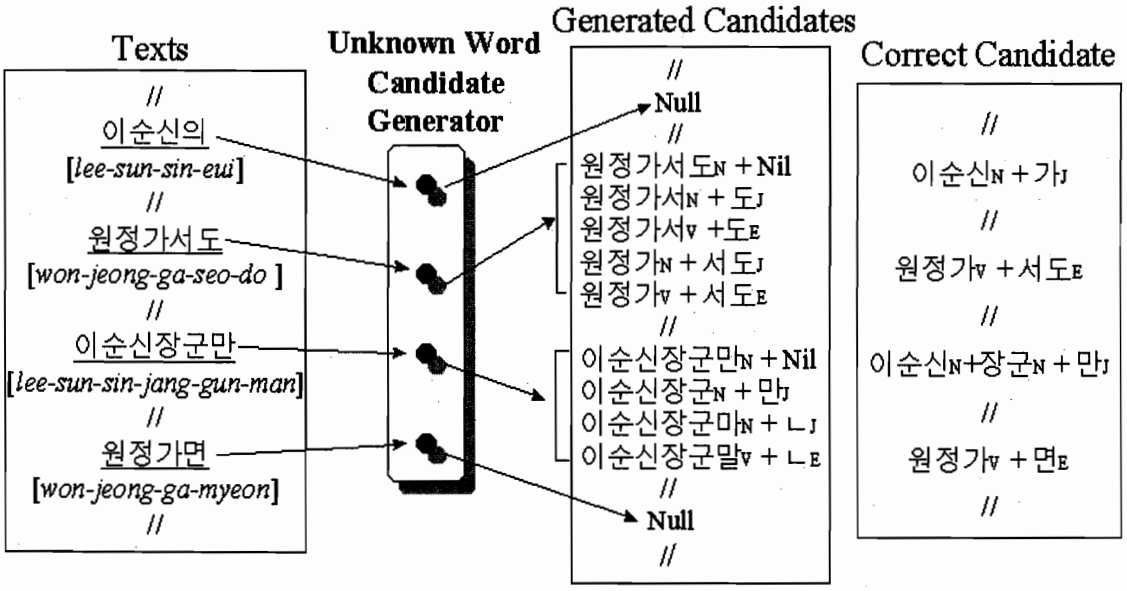


Figure 3. The existing unknown noun recognizing method

Our approach, named an example analysis method (Park 1997), is based on a comparative analysis of several example eojeols. We accept a candidate for an unknown word only if the candidate is consistently applied to its example eojeols.

[7] The tag 'N' stands for a noun and the tags 'J' and 'E' stand for a josa and an eomi respectively. In this case, an eomi is a tail combined with a verb head in Korean.

[8] The eojeol 이순신의[*lee-sun-sin-eui*] should be analyzed into 이순신$_{N}$+의$_{J}$, but it is improperly analyzed into 이순$_{N}$+신의$_{N}$, and the eojeol 원정가면[*won-jeong-ga-myeon*] should be analyzed into 원정가$_{V}$+면$_{E}$, but it is improperly analyzed into 원정$_{N}$+가면$_{N}$.

Figure 4 shows the basic idea of the example analysis method. The example analysis method comparatively analyzes the example eojeols 이순신의[*lee-sun-sin-eui*] and 이순신장군만[*lee-sun-sin-jang-gun-man*] and then recognizes the unknown noun 이순신[*lee-sun-sin*] uniquely. Also, this method comparatively analyzes the example eojeols 원정가서도[*won-jeong-ga-seo-do*] and 원정가면[*won-jeong-ga-myeon*] and then recognizes the unknown verb 원정가[*won-jeong-ga*] uniquely.

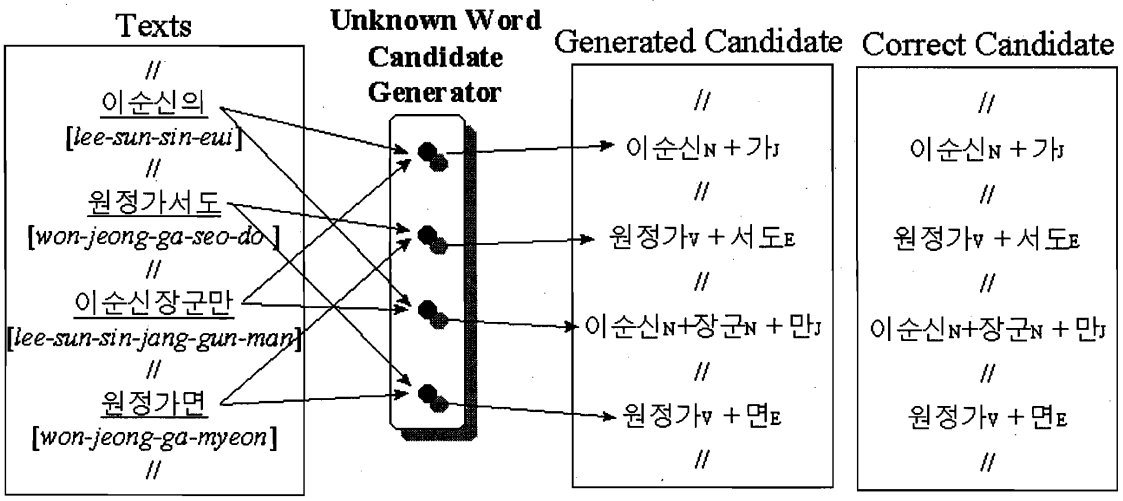


Figure 4. The example analysis method

Moreover, the example analysis method has a good effect especially on recognizing unknown proper nouns by using their lexical clues because it can effectively collect all distributed lexical clues of each unknown proper noun. Many proper nouns occur more than two places in their source text. Thus, two or more different lexical clues can appear together with an unknown proper noun. Our proper noun recognition method uses every distributed lexical clues to recognize unknown proper nouns.

4. Recognizing Unknown Proper Nouns

As mentioned in the section 2, the method of extracting lexical clues is to construct the set of heads including proper noun candidates(X) and the set of all heads(A), and then regard the known word(c) as a lexical clue for recognizing the proper noun when the occurrence probability of the known word(c) in the set(X), P(c|X), is larger than the occurrence probability of the known word(c) in the set(A), P(c|A). But, all the extracted lexical clues don't have equal clue powers because the P(c|X), P(c|A) and P(c|X)/P(c|A) can be different among the lexical clues. Therefore, it is necessary

to estimate clue powers of each lexical clue and use these lexical clues differently according to their clue powers.

The clue power is estimated based on the following two assumptions. The first assumption is that a lexical clue(c) with high probability P(c|X) guarantees the recognition of many proper nouns. And the second assumption is that a lexical clue(c) with high value P(c|X)/P(c|A) guarantees the exact recognition of proper nouns. Therefore, the clue power(CPower, henceforth) is defined as follows:

$$\text{CPower}(c) = P(c|X) * \frac{P(c|X)}{P(c|A)} = \frac{P(c|X)^2}{P(c|A)} \quad (1)$$

Equation (1) deals with only one lexical clue, but an unknown noun can have two or more distributed lexical clues. Therefore, we need to estimate the combined clue power of two or more lexical clues. So, we extend Equation (1) to Equation (2) assuming that the combined clue power of two or more lexical clues is equal to the summation of the clue powers of the componet lexical clues[9].

$$\text{CPower}(c_1, c_2, ..., c_n) = \sum_{i=1}^{n} CPower(c_i) \quad (2)$$

For example, if lexical clues *meseum* and *curator* occur near a proper noun candidate, the combined clue power of these lexical clues for this proper noun candidate is as follows:

$$\text{CPower}(\text{ museum, curator }) = \text{CPower}(\text{ museum }) + \text{CPower}(\text{ curator })$$

By using two or more lexical clues together, even lexical clues with low clue powers[10] can be used in recognizing unknown proper nouns.

After we estimate the clue power of lexical clues of an unknown noun, we decide whether the unknown noun is a proper noun or not by comparing the clue power of its lexical clues with the predetermined threshold value as shown below.

$$\text{CPower}(c_1, c_2, ..., c_n) > T$$

[9] In equation (2), c_i is the i-th lexical clue of the unknown noun.

[10] There are two cases that a good lexical clue has a low clue power. The first case is that a data sparseness problem causes the factor P(c|X) of the clue power to be low. And the second case is that a lexical clue word has multiple meanings and only one meaning of them implies a clue. This case causes the factor P(c|X)/P(c|A) of the clue power to be low. However, these lexical clues are also used to recognize unknown proper nouns if two or more lexical clues are used together and their combined clue power is above the threshold.

Determination of a threshold value is an important factor to recognize unknown proper nouns. In this paper, we determine the threshold value based on the recall rate. In an ideal case, all lexical clues extracted from the set of heads including proper noun candidates(X) guarantees 100% recall rate. This percentage corresponds to the summation of occurrence probabilities of all the lexical clues over the set(X). That is, we can say each lexical clue affects the increment of the recall rate(R) by its occurrence probability P(c|X). Therefore, we determine the appropriate threshold(T) according to the required recall rate(R) as follows:[11]

$$T = \text{CPower}(C_k) \quad \text{where } \sum_{i=1}^{k} P(c_i|X) > R \text{ and } \sum_{i=1}^{k-1} P(c_i|X) \leq R \qquad (k \leq N)$$

$$P(c_i|X) \geq P(c_j|X) \quad (i > j)$$

According to this formula, the threshold is decided to be CPower(c_k) when lexical clues(c_i) are sorted in the descending order according to their ccurrence probabilities, P(c_i|X), and the summation of P(c_1|X) to P(c_k|X) is above the recall rate, but the summation of P(c_1|X) to P(c_{k-1}|X) is not above the recall rate.

5. Experiment

5.1. Extraction of lexical clues

We extracted 274,682 unique eojeols which fail to be morphologically analyzed from 10 million eojeol size corpus. From those eojeols, we excluded eojeols having affix-derived words and spacing errors, and constructed the set(X) with the unique heads of the remaining eojeols. And then, we selected 5,486 lexical clue candidates from the set(X) and estimated their occurrence probabilities in the set(X). Also, we constructed the set(A) with 563,057 unique heads of the entire corpus and estimated the occurrence probabilities of the lexical clue candidates in the set(A). And we acquired 1,416 lexical clues by comparing the occurrence probabilities of the lexical clue candidates in those sets(X and A). The CPowers of 503 lexical clues among them are above the threshold with 80% recall rate[12]. The table 1 shows the example of lexical clues extracted with high clue powers.

---

[11] N is the number of all lexical clues.

[12] This means that unknown nouns with such a lexical clue in their neighbor are recognized as unknown proper nouns. The other lexical clues are used to recognize unknown proper nouns only if two or more lexical clues are used together and their combined clue power is above the threshold.

Table 1. The example of lexical clues

Lexical clue	Meaning	Lexical clue	Meaning
의원[*eu-won*]	member	선생[*seon-saeng*]	teacher
장관[*jang-gwan*]	minister	출신[chul-sin]	affiliation
대표[*dae-pyo*]	delegate	그룹[*geu-rup*]	business group
변호사[*byeon-ho-sa*]	lawyer	아파트[*a-pa-t*]	apartment
총리[*chong-lea*]	premier	사장[*sa-jang*]	president of company
부장[*bu-jang*]	manager	지역[*ji-yeok*]	district
박사[*bak-sa*]	doctor	대학[*dae-hak*]	university
병원[*byeon-won*]	hospital	정권[*jeong-gwon*]	regime
교수[*gyo-su*]	professor	대변인[*dae-byeon-in*]	spokesman
대통령[*dae-tong-lyeong*]	president	위원[*wi-won*]	committee
총장[*chong-jang*]	president of university	은행[*eun-haeng*]	bank
검사[*geom-sa*]	prosecutor	백화점[*baek-hwa-jeom*]	department store

5.2. Recognition of proper nouns

To verify the proposed method, we used 120,000 word size test corpus collected from newspapers and novels. Figure 5 shows the distribution of unknown proper nouns within the test corpus and Table 2 shows the comparison between the josa splitting method and the proposed method[13].

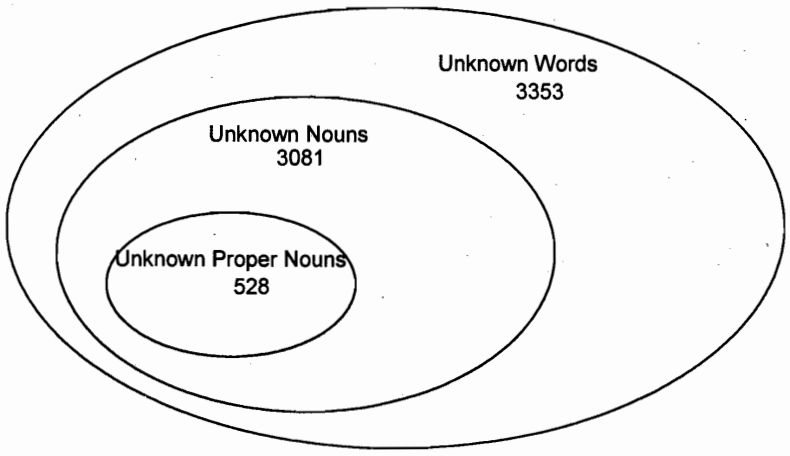


Figure 5. Distribution of unknown proper nouns

[13] The threshold applied to the proposed method of recognizing unknown proper nouns is determined when the expected recall rate is 80%.

Table 2. The comparison between the josa splitting method and the proposed method

Item	Josa splitting method		Proposed method	
	recall	precision	recall	precision
Unknown nouns	76.2	87.0	86.3	94.7
Unknown proper nouns	90.2	20.3	71.7	92.4

According to Table 2, the josa splitting method is superior to the proposed method in terms of the recall rate, but the proposed method is much superior to the josa splitting method in terms of precision. And, in Table 2, the recall rate of the josa splitting method may be expected to be almost 100% because this method regards all noun candidates as proper nouns, but it turns out to be only 90.2% because the josa splitting method can't recognize unknown proper nouns from improperly analyzed eojeols and such eojeols are 9.8% of all eojeols including unknown proper nouns. And the precision rate of the josa splitting method is very low because only about 20% of all unknown nouns were proper nouns at least in this test corpus. On the other hand, the precision rate of the proposed method is very high. And moreover, the proposed method can recognize unknown proper nouns from improperly analyzed eojeols and split an unknown proper noun and known words in the same eojeol.

6. Conclusion and Future Work

In this paper, we have presented a method of recognizing unknown proper nouns by using automatically extracted lexical clues. Our method consists of two stages. The first stage is to extract stochastically lexical clues which prefer to occur with proper nouns. And the second stage is to recognize unknown nouns having one or more lexical clues in their neighborhood and determines whether the unknown nouns are proper nouns or not by applying the given threshold to the clue power of those lexical clues. Experimental result shows that our method extracts 1,416 lexical clues from about ten million word size raw corpus, and recognizes unknown proper nouns in 92% precision rate and 72% recall rate respectively.

In the future work, we will cluster the selected lexical clues by Kohonen's SOFM(Self-Organizing Feature Map) (Pandya 1996) to recognize unknown proper nouns according to their categories. And, we will try to extend the scope of

extracting lexical clues to the adjacent eojeols which are located near the eojeols including unknown proper noun candidates.

Reference

Atwell, Eric, Stephen Elliott, "Dealing with ill-formed Englisth text," *The computational Analysis of English: a corpus-based approach*, Longman, 1987, pp.120-138.

Lee, Sang-Ho, et al, "A Korean part-of-speech tagging system with handling unknown words," *Proc. of 1995 International Conference on Computer Processing of Oriental Languages*, Nov. pp.23-25, Honolulu.

Mikheev, Andrei, "Unsupervised Learning of Word-Category Guessing Rules," *Proc. of the 34th ACL*, 1996, pp.327-334.

Pandya, Abhijit S. and Robert B.Macy, *Pattern Recognition with Neural Networks in C++*, CRC Press, 1996.

Park, Bong-Rae, Hae-Chang Rim, "A Korean Corpus Refining System based on Automatic Analysis of Corpus," *Proc. of Natural Language Processing Pacific Rim Symposium*, 1995, pp.89-94.

Park, Bong-Rae, Young-Sook Hwang, Hae-Chang Rim, "Recognizing Korean Unknown Words by Comparatively Analyzing Example words," Proc. of 1997 *International Conference on Computer Processing of Oriental languages*, pp.127-132, Hong Kong.

Strzalkowski, Tomek and Jin Wang, "A Self-Learng Univeral Concept Spotter," In *Proceedings of COLING-96*, 1996, pp.931-936.

Weischedel, Ralph, Marie Meteer, Richard Schartz, Lance Ramshaw, "Coping with Ambiguity and Unknown Words through Probabilistic Models," *Computational Linguistics*, Vol.19, 1993, pp.360-382.

Yang, Jang-Mo, Min-Jung Kim, Hyuk-Chul Kwon, "Extraction Method of the Unknown-Words with Linguistic Knowledge in Korean," *Proc. of Spring Conference of Korean Information Science Society*, 1996, pp.925-928. (in Korean)

Logical Operators and Quantifiers in Natural Language

Shin-ichiro KAMEI and Kazunori MURAKI

C&C Media Research Laboratories, NEC Corporation
4-1-1, Miyazaki, Miyamae-ku, Kawasaki, 216 JAPAN
kamei@hum.cl.nec.co.jp, k-muraki@hum.cl.nec.co.jp

abstract

This paper investigates negations in natural language, comparing natural language and first-order logic, and it introduces a model for describing quantifiers and negations in natural language. The model consists of the semantic representation of quantifiers such as 'all' and 'some,' and logical operators such as entailment between these words and thier negation. The basic framework of semantic representation is a pair of lists, which is called a 'dual list.' Each list includes conceptual elements. The upper list represents a 'literal' meaning of a word, and the lower represents a 'possible' meaning. Logical operations such as entailment and negation are defined on the dual list. The model can handle a wide range of linguistic phenomena, which are related to numbers such as 'three,' and conjunctions such as 'and' and 'or,' as well as qualities such as 'all' 'some' and 'no.' The negation operation on the dual lists consistently generates all possible interpretations for these words. Words such as 'all' and 'some,' and 'and' and 'or' correspond to logical operators in a sense, but they are different in some other aspects. This model, especially its negation process, clarifies the similarities and differences between logic and natural language.

1 Introduction

In natural language, there are many curious phenomena which are difficult to explain from the viewpoint of usual first-order logic. Let us consider the following sentence, which includes a numeral, as an example.

(1) I solved three of the problems.

A natural interpretation of this sentence is "I solved just three of the problems, not all or four or two or one or none of them." However, in a logical way, this statement is true, when "I solved FOUR of them." For example, if the border line between success and failure of a test is three, this sentence is naturally spoken, even when, in fact, the person solved four of the problems (Chomsky, 1972; Ota, 1980; Ikeuchi, 1985). This phenomenon suggests that more complex states than just 'three' for the meaning of the number three are needed to understand natural language.

The following is a Yes/No question corresponding to sentence (1) and its answers. Interestingly, both of the answers below are possible in this case (Ota, 1980; Ikeuchi,

1985). The fact that both Yes and No answers are possible for the same situation suggests the 'duality' of number concepts.

(2) A: Did you solve three of the problems ?
B: – Yes, in fact I solved four.
– No, I solved four.

In addition, let us think of a negative sentence which corresponds to sentence (1). It is well known that a negative sentence like this can have several interpretations.

(3) I didn't solve three of the problems.

One possible interpretation of sentence (3) is that there are three problems that "I did not solve." Another interpretation of this sentence is that some of the problems were solved, but that the number did not reach three. In addition, an interpretation that the number of solved problems exceed three is also possible. What is important here is that these interpretations are all related to the negation of the number. This phenomenon also suggests the concept 'three' in natural language has to be more complex than just the number 'three' in mathematics.

Similar interesting phenomena are known for quantifiers such as 'all' and 'some,' and conjunctions such as 'and' and 'or.' These words in a sense are similar to operators in logic, that is, '$\forall$' '$\exists$' '$\wedge$' and '$\vee$'. However, they are definitely different in many other cases, especially in negation processes.

The facts described above have already been pointed out by previous research but these phenomena have not been treated satisfactorily. This paper compares natural language and logic, and introduces a model to describe phenomena which are related to the words above and negations in natural language. The model describes, explains, and calculates the possible interpretations of a wide range of affirmative, interrogative, and negative sentences in a consistent and visible way, and clarifies the similarities and differences between natural language and logic.

2 The Dual List Model

2.1 Dual List Expression

This section introduce the basic conceptual representation of the model, i.e. the dual list. This representation is introduced for number concepts, but is applied to a wide range of concepts such as 'all' and 'some,' and 'and' and 'or.'

As we have seen in the previous section, the number part of sentence (1) seems to be expressed by more complex representations than just the mathematical number.

(1) I solved three of the problems.

The authors think that five states are actually needed for clarity: (i) All problems are solved, (ii) the number of solved problems exceeds the number in the sentence (=three in this case), (iii) the number of solved problems is exactly the number in the sentence, (iv) the number of solved problems does not total the number in the sentence, and (v)

no problems are solved. The authors introduce five primitives, 'A,' '>n,' '=n,' '<n,' and 'N,' which are abstracted from the above five states.

In order to describe the actual meanings of these states, a list of these primitives is used. The five primitives are arranged in a list.

(4) {A, >n, =n, <n, N}

The five states are represented by the relative positions shown in Table 1. The authors think that the meanings of words are identified by their relative positions in the list expression. In these lists, '-' means that the value in that particular position is lacking.

Table 1: five states for expressing a number

States	{A,	>n,	=n,	<n,	N}
all	{A,	−,	−,	−,	−}
>three	{−,	>n,	−,	−,	−}
three	{−,	−,	=n,	−,	−}
<three	{−,	−,	−,	<n,	−}
none	{−,	−,	−,	−,	N}

The fact that the two answers with Yes and No are both possible for the same situation, exemplified by sentence (2), suggests that the number concept in sentence (1) has a kind of duality. The authors represent the meaning of the number using the following dual list.

(5) $\begin{Bmatrix} -,-,=n,-,- \\ A,>n,=n,-,- \end{Bmatrix}$

The upper row (the direct 'literal' meaning in this representation shows the state where the number of solved problems is the number in sentence (1). The lower row (the possible interpretation) expresses the possible numbers of solved problems, when sentence (1) is spoken. For example, this statement is false, when the number of the solved problems is TWO. Logically, however, this statement is TRUE, when the number of solved problems is FOUR.

When this sentence is spoken, its number part conveys meanings which correspond to BOTH the rows in the dual list. That is, meaning is not only indicated by the upper 'direct' row, but also by the lower 'possible' row.

2.2 Intersection Operation

This section introduces an intersection operation on the list, and it shows that dual list and intersection operation naturally generate the two possible answers for the same situation, described in utterance (2).

In an affirmative sentence, the upper 'direct' meaning may be dominant. However, in the case of an interrogative sentence, the lower 'possible' meaning plays a more important role. This model explains the two possible answers for utterance (2) in a simple way. In

Fig. 1, the meaning of the question is expressed with a dual list. The meaning of the real situation (the meaning of 'four' here) is expressed with a single list (in the middle), because it is not an interpretation, but a situation. When comparing the upper row of the question and the row expressing the situation 'four,' there is no common value. There is no intersection between them. This corresponds to the answer with 'No.' When comparing the lower 'possible' row and the situation, there is an intersection, that is, the value '>n.' Therefore the answer is 'Yes.' This intersection operation is a simple and natural way to calculate possible answers to a question which includes a number.

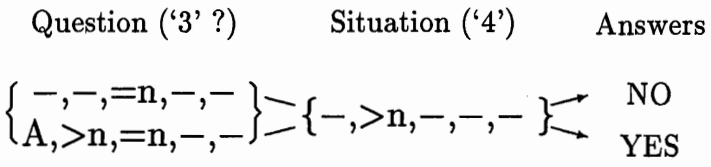


Figure 1: Intersection Operation for Q and A

The difference between the literal meaning and the implications of an utterance is called Conversational Implicature (Grice, 67). The difference between the two rows, 'A' and '>n' in this case, expresses the possibilities of Conversational Implicature.

2.3 Negation Operations

This section introduces Negation Operations, which are defined on the dual list representation. Sentence (3) is a negative sentence which corresponds to sentence (1). A negative sentence like this has several interpretations which has been pointed out but which has not been dealt with treat satisfactorily. This model calculates all the possible interpretations of a negative sentence from the representation of the original affirmative sentence.

(3) I didn't solve three of the problems.

One possible interpretation of sentence (3) is that there are three problems that "I did not solve" (Interpretation A). In this interpretation, the number 'three' is not under the influence of negation, that is, the number is out of the scope of negation. To obtain this interpretation, it is not necessary to change the dual list for the original affirmative sentence (5). It is necessary to change the meaning of the values from the number of solved problems to the number of unsolved problems in the representation of the original affirmative sentence (Fig. 2). The lower row expresses the possibility that the number of unsolved problems exceed three.

Where the number (=three in this case) is within the scope of negation, the negative sentence requires other interpretations.

(6) A: Did you solve three of the problems?
B: No, I didn't (get to) solve three of the problems.

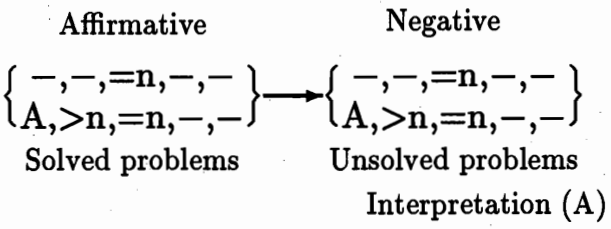


Figure 2: One Negative Interpretation from Affirmative Dual List

——— Interpretation (B)

Response B might mean that some of the problems were solved, but that the number did not reach three. This interpretation can be obtained from the model shown in Fig. 3. The negation operation is shown in Table 2.

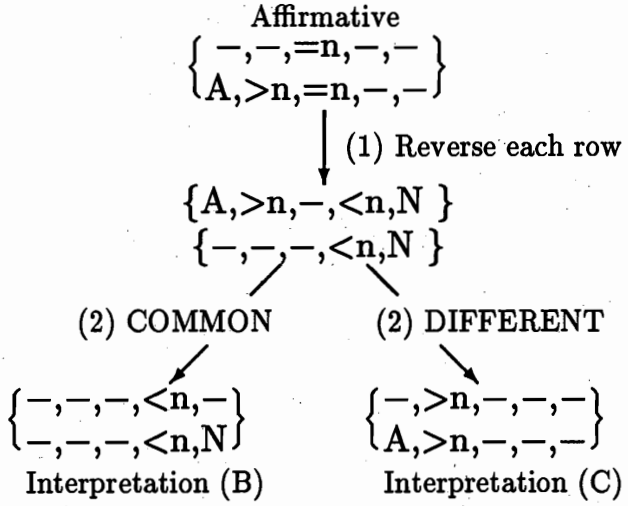


Figure 3: Two Negative Interpretations from Affirmative Dual List

1. Reverse each affirmative row.
2. Select the COMMON part of the two rows.
   The result is a new possible interpretation row.
3. Omit the edge values (A and N).
   The result is a new direct meaning row.

Table 2: Negation Operation for Interpretation B

Step 1 in Table 2 realizes a primitive negation operation on each row. This interpretation of the negative sentence is consistent with the negations of both the direct meaning

and possible implications. Step 2 realizes this condition. This interpretation usually implies that there are some solved problems. This means that negation usually does not deny the existence of solved problems. However, in a logical way, no problem being solved is a possible situation. Step 3 realizes this condition.

(7) A: Did you solve three of the problems?
C: No, I didn't solve THREE of the problems: I solved ALL of them.

——— Interpretation (C)

The above is a possible utterance, which requires another interpretation. Table 3 shows the procedure to calculate this interpretation (Interpretation (C)).

1. Reverse each affirmative row.
2. Select the DIFFERENT part of the two rows.
   The result is a new possible interpretation row.
3. Omit the edge values (A and N).
   The result is a new direct meaning row.

Table 3: Negation Operation for Interpretation C

This interpretation differs from interpretation B, only at Step 2, that is, 'to select the DIFFERENT part of the two rows.' This means that the interpretation is consistent with only the negation of the direct meaning, and it does not satisfy the negation of the possible implications. Step 2 realizes this condition. This exemplifies that the Conversational Implicature can be canceled. In speech, stress is put on THREE and ALL in this interpretation, and this linguistic phenomenon is accounted for in Step 2.

3 Quantifiers in Natural Language

3.1 'All,' 'no,' 'some,' and 'not all'

Here, we will apply the same model introduced in the previous section to the relations between 'all,' 'some,' 'no,' and 'not all' in natural language.

It is well known that sentence (8-1) logically entails sentence (8-2). Sentence (8-2) usually implies sentence (8-3). However, sentence (8-3) contradicts the original sentence (8-1). A careless mixture of logical implication and usual implication in language makes the inference of (8-3) from (8-1) unreasonable (Horn, 1972; Ota, 1980; McCawley, 1981).

(8-1) All students are intelligent.
(8-2) Some students are intelligent.
(8-3) Some students are NOT intelligent.

The discrete list model is a useful tool for describing these relations. List (9) is used to express relations between 'all,' 'some,' 'no,' and 'not all (= some ... not).' In this case, three primitives, 'A' 'S' and 'N' are used. These primitives are abstracted from the

meanings of 'all' 'some' and 'no,' respectively. In this list, the value 'S' corresponds to the state wherein there are SOME students who are intelligent and SOME other students who are NOT intelligent.

(9) {A, S, N}

The meanings of these words are also expressed with a dual list. Figure 4 graphically and simply represents the complicated relations among the words.

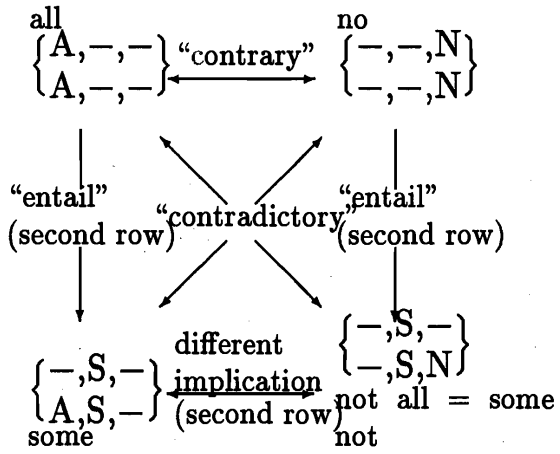


Figure 4: 'All,' 'some,' 'no,' and 'not all'

In Fig. 4, the second 'possible' rows for 'all' and 'some' have an intersection at the value 'A.' 'No' and 'not all' have a similar intersection. This realizes entailment between the two concepts. Figure 4 also expresses the difference between 'contrary' and 'contradictory.' If 'all' is true,'no' is false. If 'no' is true, 'all' is false. Both expressions cannot be true at the same time. However, these two CAN BE FALSE at the same time, because it is possible that some students are intelligent and some students are not. The term 'contrary' expresses this relation. On the other hand, 'all' and 'not all' have a different relationship. These two cannot be true at the same time, and cannot be false at the same time. 'No' and 'some' have the same constraint. The term 'contradictory' in Fig. 4 expresses this relation.

3.2 Negation of Quantifiers

An important point here is that the same operation of negation, Table 2, used for numbers can also obtain the representation of 'not all' from that of 'all' in Fig. 4. The other negation operation, Table 3, produces nothing in this case (Fig. 5). The negation operations are basic and general.

The word 'all' is similar to the operator '∀' in logic, and the word 'some' is similar to '∃'. However, the relations concerning the words 'all' and 'some' in natural language are more complicated than the relations between the two operators in logic.

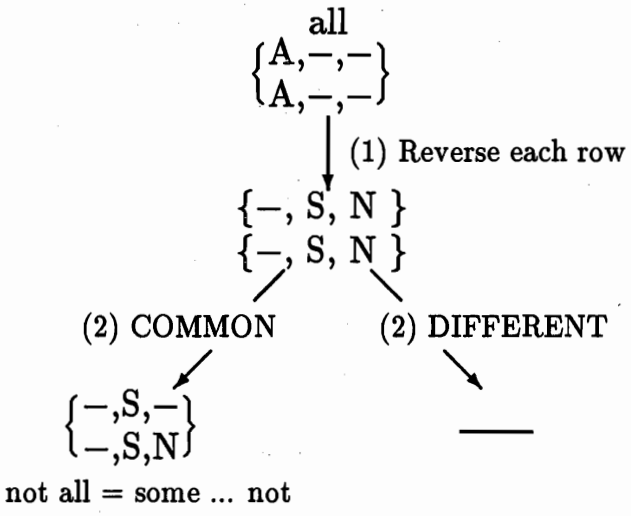


Figure 5: Negation Operation executed on 'ALL'

For example, in the case of logic, $\neg\ (\forall\ A) = \exists\ \bar{A}$, and $\neg\ (\exists\ A) = \forall\ \bar{A}$. These two relations are symmetric.

It is true that the negation of the word 'all' is 'some ... not,' as described above. Natural language and logic are similar at this point. However, this is not the case with the word 'some' and the operator '$\exists$'. The negation of 'some' is difficult to consider in natural language.

The dual list is able to explain this phenomenon. Figure 6 shows negation operations on the word 'some.' Both 'common' and 'different' results have a vacant list as an upper 'direct' meaning. This corresponds to the fact that the negation of 'some' is difficult to consider, while the negation of the logical operator '$\exists$' is easy. The model, that is the dual list, the intersection operation, and the negation operation, is useful and powerful.

4 Operators 'AND' and 'OR'

4.1 'OR' in Natural Language and 'OR' in Logic

This section applies the same model to the conjunctions 'and' and 'or' in natural language. These words are similar to the logical operators '$\wedge$' (='AND') and '$\vee$' (='OR'). However, natural language and logic are definitely different. The dual list model clarifies the similarities and differences between natural language and logic.

It has been shown that 'OR' has characteristics similar to degree concepts such as numbers, 'all' and 'some' (Gazdar, 1979). The fact that 'or' in natural language generally has two interpretations, the 'inclusive or' and the 'exclusive or' suggests that the concept of 'or' can be expressed by the dual list.

To express concepts for conjunctions such as 'and' and 'or' in natural language, the authors introduce three states: (++), (+−/−+), and (−−). These primitives are abstracted from the meanings of 'and' 'exclusive or' and 'nor.' A basic list is as follows.

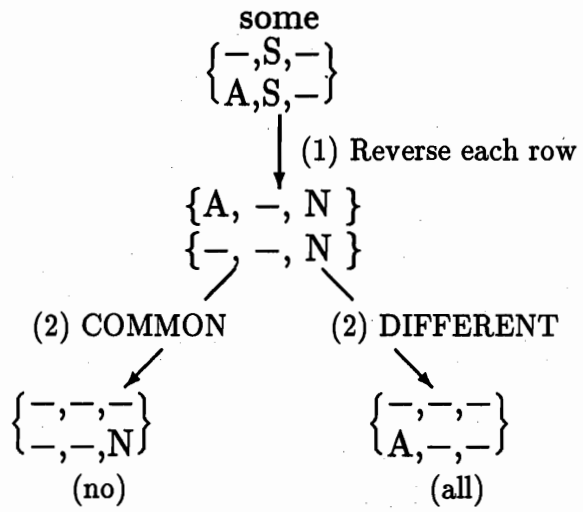


Figure 6: Negation Operation executed on 'SOME'

(10) $\{(++), (+-/-+), (--)\}$

The dual lists for 'and' and 'or' are as follows.

(11) and $\left\{\begin{matrix}(++), & - & , & - \\ (++), & - & , & - \end{matrix}\right\}$

(12) or $\left\{\begin{matrix} - & ,(+-/-+), & - \\ (++),(+-/-+), & - \end{matrix}\right\}$

'Exclusive or' is a direct meaning of 'or' and 'inclusive or' is a possible interpretation of 'or' in this framework. While 'or' in logic is usually 'inclusive or,' the authors treat 'or' in natural language as the dual list above. In other words, 'or' in natural language means BOTH 'exclusive or' (as a direct or literal meaning) and 'inclusive or' (as a possible meaning).

4.2 Negations of 'and' and 'or' in Natural Language

This section calculates the negations of 'and' and 'or' in natural language. Logically, the negation of the logical operation 'OR' (that is, 'Inclusive or') is 'NOR.' However, in a sense in natural language, 'AND' instead of 'NOR' can also be a negation of 'OR.'

It is difficult to conceptualize the negation of 'or' in natural language, in the usual sense, although the negation of 'and' is easy. Figure 9 shows the relationship between the inclusive and exclusive 'or' and their negations.

The same negation operations will produce the two negations for 'or,' that is, both NOR and AND. The direct meaning rows in the two interpretations of negations for 'or'

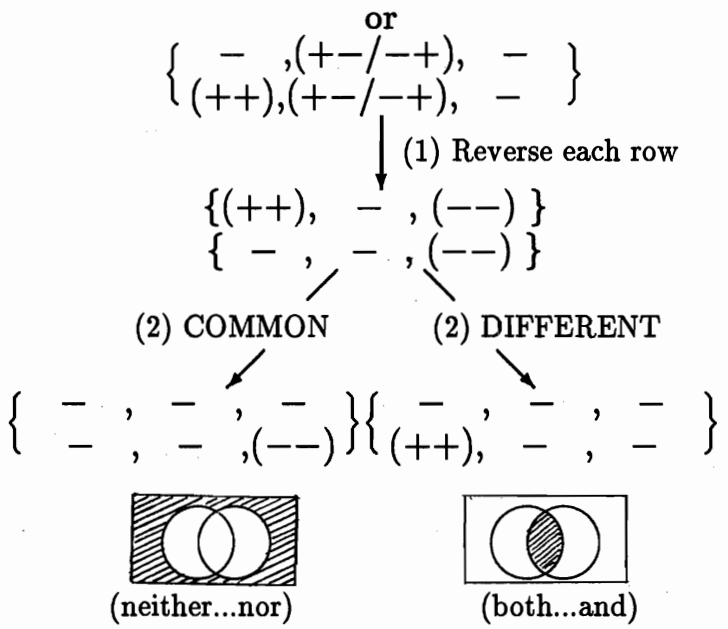


Figure 7: Negation Operation executed on 'OR'

have no values. This corresponds to the fact that it is difficult to consider the negation of 'or' in natural language. Note that the dual list for 'or' and the dual list for 'some' in Fig. 4 have an identical structure. It is equally explained that the negation of 'some' is difficult to conceptualize in natural language, while the negation of 'all' is easy.

5 Conclusion

This paper presented a model for negations in natural language. The characteristics of the model are: (1) discrete conceptual primitives, (2) list representation of concepts, (3) dual list representation for possibilities of Conversational Implicature, (4) intersection operation on the list for realizing entailment of two concepts, and (5) negation operations on the dual list to calculate all the possible interpretations of negation of concepts.

The model describes, calculates, and explains a wide range of linguistic phenomena, such as: (1) All possible answers to a question which contains a quantitative word, (2) all possible interpretations of negation of quantitative words, (3) the difficulty of applying negation to some quantitative words, such as 'some' and 'or,' and (4) the relations between 'OR' in logic and 'or' in natural language. These phenometa suggest that the model is able to represent substantial structures in natural language and that it is a suitable tool for natural language understanding.

Since the model uses conceptual elements, and concepts are defined by relative positions using the list, the model can easily be applied to other quantifiers such as 'many,' 'few,' and 'a few' (Kamei and Muraki, 94). The authors hope that this model will become a possible extension of first-order logic for natural language understanding.

References

[1] Barwise, J. and R. Cooper (1981). 'Generalized quantifiers and natural language.' *Language and Philosophy 4.2*, pp. 159-219.

[2] Bolinger, D. L. (1972). *Degree Words*. Mouton.

[3] Chomsky, N. (1972). *Studies on Semantics in Generative Grammar*. Mouton.

[4] Fauconnier, G. R. (1975). 'Pragmatic scales and logical structure.' *Linguistic Inquiry 6*, pp. 357-375.

[5] Gazdar, G. (1979). *Pragmatics: Implicature, Presupposition, and Logical Form*. Academic Press.

[6] Grice, H. P. (1975). 'Logic and conversation' In P. Cole and J. L. Morgan Eds., *Speech Acts*, pp. 45-58. Syntax and Semantics 3. Academic Press.

[7] Hirschberg, J. B. (1985). 'A Theory of Scalar Implicature' Ph. D. dissertation, University of Pennsylvania.

[8] Horn, L. R. (1972). 'On the semantic properties of logical operators in English' Reproduced by the Indiana University Linguistics Club (1976).

[9] Ikeuchi, M. (1985). *Meishi-ku no Gentei Hyougen (Noun Phrase Specifying Expressions)*, (in Japanese). Taishukan.

[10] Kamei, S. and K. Muraki (1988). On a Model of Degree Expression, (in Japanese). *NLC 88-6*. The Institute of Electronics, Information and Communication Engineers.

[11] Kamei, S., A. Okumura, and K. Muraki (1990). Syntax of English Adverbs, (in Japanese). *Proceeding of the 40th Conference of Information Processing Society of Japan, Vol. 1*, pp. 417-418.

[12] Kamei, S., and K. Muraki (1994). A Discrete Model of Degree Concept in Natural Language. *Proceeding of the 15th International Conference of Computational Linguistics, Vol. II*, pp. 775-781.

[13] McCawley, J. D. (1981). *Everything that Linguists have Always Wanted to Know about Logic but were ashamed to ask*. The University of Chicago Press.

[14] Ota, A. (1980) *Hitei no Imi (Meanings of Negation)*, (in Japanese). Taishukan.

[15] Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik. (1985). *A Comprehensive Grammar of the English Language*. Longman.

[16] Yagi, T. (1987). *Teido Hyougen to Hikaku Kouzou (Degree Expressions and Comparative Structures)*, (in Japanese). Taishukan.

Chinese text Compression using Chinese Language Information Processing

Jun Gao **Xixian Chen**

Beijing University of Posts and Telecommunications 103 Box
No. 10, Xi Tu Cheng street, Hai Dian district, Beijing, 100088, China
e-mail: b9507311@bupt.edu.cn

Abstract

To transport the large scale authentic Chinese text, especially in the personal telecommunication system, it is necessary to establish a highly efficient coding method. A good coding system can reduce transportation time. There are two basic ways to compress information. One way is to get rid of the perplexity is in the relativity of information source. The other is to remove the perplexity of not equal probabilistic distribution. In this paper, a novel information compressing method is presented. It utilizes the relativity of the information source and catchs the information of different probabilistic distribution according to the definition of Chinese words. The relativity and not equal probabilistic distribution are connected by a optimum searching method. The aim of high compressing ratio is reached. And. Some experimental results are also covered.

Keywords: **Chinese corpus, optimum searching, Huffman coding, Discrete Stable Information Source, Maximum likelihood, relativity, perplexity, entropy, average code length.**

中文信息处理中的信息压缩

高军　　陈锡先

北京邮电大学信息技术实验室
北京市海淀区西土城路10号北京邮电大学103信箱
北京，100088，中国
e-mail: b9507311@bupt.edu.cn

摘要

在传送大规模真实中文信息中，特别是在个人通信系统中，准确高效的信息压缩方法是极为必要的。一个好的信息压缩方法，可以节约大量的传输时间和存储空间。信息压缩有两种基本途径，第一种是，去处寓于信源的相关性之中冗余度，另一种是，去处寓于概率的非等分布之中的冗余度，改变信源的概率分布，以期尽可能达到等概率分布的目的。本文提出了一种全新的信息压缩方法，这种方法在理论上充分利用了信源的相关性，同时又根据中文信息处理中对词的界定，把握了其不等概分布特性，运用最优搜索方法，把信源相关性与信源的不等概分布有机地结合起来，从而达到了高效压缩信息的目的。在文中，列举了实例结果。

关键词　汉语语料　最优搜索　Huffman 编码　最大似然度　相关性　冗余度　熵　平均码长

1. 引言

在通信系统中，采取准确高效的信息压缩技术，对于解决存储容量的限制、信道带宽的不足，具有重大意义。尤其是在传输大量中文字符信息的过程中，信息在压缩后应能够完全恢复，这对于压缩技术提出了更高的要求。

数据的信息压缩有两种基本途径，第一种是，对于联合信源，其冗余度寓于信源的相关性之中，去处他们的相关性，使之成为或差不多成为不相关的信源。另一种是，对于离散无记忆信源，其冗余度则寓于概率的非等分布之中，改变信源的概率分布，以其尽可能达到等概率分布的目的。

具体到信息压缩编码技术上则表现为两种不同的处理方式：一种是信息被压缩，在一定程度上消除了部分冗余信息，但恢复后得不到信息的完全复原，即部分有用信息也被压缩掉了，是有损压缩；如声码器(Vocoder)对语音信息的编码压缩，还有差分脉冲编码调制(DPCM)对图像信息的编码等。它们都是从模拟信号转变成数字信号的过程中对信息进行压缩的。这两种信息压缩编码得到的恢复语音和图像信号，虽然主要信息成分被保留了，但一些有用的信息也失去了，从而导致信息的失真。另一种是

缩后的还原信息应能够百分之百地得到恢复。即信息压缩和还原不应带来任何的损失。中文信息处理中的信息压缩就属于这种字符信息的无损压缩。

为了节省信息存储空间和提高通信效率，应尽可能把这些信息的冗余成分去掉。在文件和通信信息中，冗余信息主要有：字符分布冗余，即，少数字符的使用频度明显高于其他多数字符；高频组合冗余，既有些组合字符比其它的字符更常见，这些冗余信息的存在，为字符信息的压缩提供了可能。

本文以信息理论为基础，充分利用以上提到的各种可能的冗余性，对中文信息进行高效压缩。

2. 原 理 及 算 法

2.1 理论基础

我们把中文信息作为信源来研究。根据由信息理论关于信源的论述，有以下的结论：

对于离散信源 X，若其输出的随机序列 $x_1, x_2, x_3, \cdots, x_i, \cdots$ 各维联合概率分布均与时间起点无关，即，当 $t = i, t = j, (i, j$ 为任意的整数，且 $i \neq j)$ 时，有：

$$P(x_i) = P(x_j)$$

$$P(x_i x_{i+1}) = P(x_j x_{j+1})$$

$$\vdots$$

$$P(x_i x_{i+1} \cdots x_{i+N}) = P(x_j x_{j+1} \cdots x_{j+N})$$

那么，具有这样性质的离散信源是完全平稳的离散信源，简称为平稳信源。

而长为N的信源符号序列中平均每个信源符号所携带的信息量，即平均符号熵为：

$$H_N(\mathrm{X}) = \frac{1}{N} H(\mathrm{X}_1 \mathrm{X}_2 \cdots \mathrm{X}_N) \qquad \cdots\cdots \quad (1)$$

平稳信源 X 的信息熵 $H(\mathrm{X})$ 与平均符号熵 $H_N(\mathrm{X})$ 的关系为

$$H(\mathrm{X}) = H_\infty = \lim_{N \to \infty} H_N(\mathrm{X}) \qquad \cdots\cdots \quad (2)$$

H_∞ 为平稳信源的极限熵。

$H(\mathrm{X})$ 可由 $H_N(\mathrm{X})$ 来近似。N的值越大，$H_N(\mathrm{X})$ 越能更好的描述 $H(\mathrm{X})$。

所以，我们可以用M组最大长度为N的信源符号序列来描述 $H(\mathrm{X})$。即，

$$H(X)\approx\frac{1}{\sum_{i=1}^{M}N_i}H\{(X_{1,1},X_{1,2},\cdots,X_{1,N_1}),$$
$$(X_{2,1},X_{2,2},\cdots,X_{2,N_2}),\cdots,$$
$$(X_{M,1},X_{M,2},\cdots,X_{M,N_M})\}\quad\cdots\cdots\quad(3)$$

(3)式中，在M个组之间，信源符号彼此是相互独立的；而在M个组的内部，N_i个信源符号之间是相互依赖的，其中$1\le N_i\le N$，$i=1,\cdots,M$。

我们根据以上分析，把中文信息模型化为特定汉字字串内部是彼此相关联的，而字串之间则是相互独立的。

这样模型化的目的是既能够利用汉字字符之间的相关信息，又能够利用汉字字串的频率信息。如何把这些相关信息及频率信息有机的结合于信息压缩中，是提高压缩效率的关键。在通常的对中文进行编码过程中，都采用已知码本对中文信息进行穷搜索、穷匹配，但这样所得的结果不是全局最优的，甚至不是局部最优的。

2.2 算法

针对上一节中的分析，本文提出了一种最优搜索算法，它利用汉字之间的相关性和字串的频率特性，使得在给定的范围内，匹配的结果是最佳的。

为了方便描述，我们预先作如下约定：

W：表示一个字符串。W的长度为N。

S：表示对W的一个划分。即：

$$S=s_1s_2,\cdots,s_k,\cdots,s_q\quad\cdots\cdots\quad(4).$$

共q个段。其中s_k为W中的一个子列。

$$s_k=[\omega_{i_k},\omega_{i_k+1},\cdots,\omega_{i_{(k+1)}-1}]\quad\cdots\cdots\quad(5)$$

I_s：表示对W中的q个段的每段段首的索引。即：

$$I_s=\{i_1,i_2,\cdots,i_q,i_{q+1}\}\quad\cdots\cdots\quad(6)$$

其中，$i_1=1$，$i_{q+1}=N+1$，

且 $i_1<i_2<\cdots<i_{q+1}$.

Len：表示每一子列的最大长度。

则由(4),(5),(6)式及其限制条件可推出：

$1\le k\le q$，且 $1\le i_{k+1}-i_k\le Len$.

$\{I_s\}$：表示满足条件的所有I_s的集合。

如前所述，我们假设各个子列之间是相互统计独立的。而子列内部是相关的，则最优搜索模型可定义为:

在一个句子中的所有可能的划分中，子列概率的乘积最大的那个划分——具有最大似然度的那个划分为: $I_{s\max} = \arg\Gamma(W)$，其中，

$$\Gamma(W) = \max_{\{s\}} \prod_{k=1}^{q} P(s_k)$$

$$= \max_{\{I_s\}} \prod_{k=1}^{q} P(\omega_{i_k}, \omega_{i_k+1}, \cdots \omega_{i_{(k+1)}-1}) \quad \cdots\cdots \quad (7)$$

在具体实现中，最优搜索算法由以下步骤实现:

第一步: 计算所有汉字串的出现概率，并从中挑选出其概率与其长度的乘积最大的前 *NumTop* 个字串，我们称这个乘积为其相应字符串的权值。

设 $N_{init}(s)$ 代表某长度为 $l\ (\leq Len)$ 的汉字串S在总长为N的文本W中出现的次数。并设 N_{total} 代表所有在长度上小于最大长度 *Len* 的汉字串出现次数的总和。

则: $$P_{init}(s) = \frac{N_{init}(s)}{N_{total}} \quad \cdots\cdots \quad (9).$$

其中 $N_{total} = \sum_{i=1}^{Len} N_i$，$N_i$ 代表所有长度为 i 的汉字串的个数。经过以上的统计，可以找出所有在长度上小于 *Len* 的字串的概率。

$$SetTop = \{s | s \in \underset{NumTop}{Top}(N_{init}(s) \times L(s))\} \quad \cdots\cdots (10)$$

其中，*SetTop* 表示满足式(10) 的 *NumTop* 个汉字字串的集合。

$L(s)$ 则表示汉字字串 s 的长度。

Top( ) 为求前 *NumTop* 个具有较大权值 $N_{init}(s) \times L(s)$ 的汉字字串。

第二步: 把汉字编码中所有一级和二级汉字（国家标准GB2312–80中，6763个汉字）以其频率得有先顺序加入到 *SetTop* 的汉字字串集合，使之成为有权值优先次序的码本

第三步: 基于第二步生成的码本，在长度为 *Len* 的范围内对中文文本进行最优匹配，得到分割结果，以便进行编码。

即: 设对长度为 L 的 W 的子串，则

$$\Gamma(\omega_1\omega_2\cdots\omega_L)$$

$$= \max_{1\leq i\leq Len} \{P(\omega_{L-i+1}\omega_{L-i+2}\cdots\omega_L)\Gamma(\omega_1\omega_2\cdots\omega_{L-i})\} \quad \cdots\cdots \quad (11)$$

重复第二步的迭代计算，最终可求出一个分段序列$I_{s_{max}}$，它能为长度是N的中文文本W提供最大的似然度值。

第四步 对中文文本的分割结果依其在码本中的权值，按一定的编码规则进行编码。

3. 试验及结果评价

3.1 中文文本的选择

我们采用名为《中国百家新闻报刊……1994》的电子出版物中的中文文本作为试验用语料来源，其总量为789兆字节。我们节选其中四月份的语料，共6.9兆字节，又从中随机节取若干段作为真实中文文本。本文中所引用的语料有23131个汉字。

3.2 中文文本的预处理

由于在电子出版物中存在大量的乱字符（ASCII码值160的半角字符）、全角的汉字与半角的各类字符混排以及统计句长的需要，我们对节取的文本进行了预处理。在汉语中存在着许多自然的切分标志，如标点符号等，汉字字串不能跨越这些标点而存在。经过预处理，我们把整个语料库切分为各种短句，其平均长度为12。标点符号集为{，。！；、：？}以及长度超过两个以上的空格。这样做不仅提高了分词精度而且减少了计算量。句间的分割符均以在其后的句子的长度所代替。平均句长为12。经统计，未曾发现语料中存在超出我国国家标准GB2312–80规定的一、二级汉字的范围。因此，本方法建立的码本可以适应实际要求。

3.3 字串最大长度的设定

对汉语中词的统计和研究(刘源,1994)表明汉语中二字词出现的数量和频度最高，随着词长的增加，词的频度则越来越小，而词长超出4的词的个数微乎其微。根据以上语言学的结论，我们在分割汉字字串的过程中，设定字串的最大长度为4。

3.4 编码方式

以统计为基础的编码大致的分为两大类，即等长编码和变长编码。它符合等概率最大熵条件其编码效率最低，而变长编码则与信源中元素出现概率的不均匀性紧密联系在一起。为此，我们采用最优变长编码--Huffman编码。采用Huffman编码的优点

布：它适应环境的能力较强，只要设计合理，概率分布上的细微变化，不至于严重降低编码效率。在本实验中，我们采用二元Huffman编码。

3.5 试验条件及运行速度

用Pentium 586/133 MHz、32M 内存计算机计算，操作系统为Windows NT 4.0。Visual C++ 4.0为编程语言及调试工具。在汉字字串的最大长度及真实文本的大小确定后，本算法的运行速度主要受其中 *NumTop* 值的变化的影响。程序运行时间随 *NumTop* 的变化如表一所示。

NumTop	200	500	1000	1500	2000	3000	5000	7000
时间	0.6 小时	1.8 小时	3.9 小时	7.7 小时	10.5 小时	14.9 小时	25.4 小时	34.9小时

表 一

3.6 结果及评价

在前 *NumTop* 个具有较大权值的汉字字串集合中，不同长度的字串在其中出现的比例是不同的。详见表二。

NumTop	200	500	1000	1500	2000	3000	5000	7000
一字串	57%	47.8%	37.9%	31.2667%	26.25%	19.8667%	13.88%	11.1285%
二字串	20.5%	27.6%	33.5%	32.9333%	34.85%	30.8%	28.8%	24.6429%
三字串	12%	13.6%	14.9%	19.4%	20.6%	24.6%	31.82%	29.3%
四字串	10.5%	11%	13.7%	16.4%	18.3%	24.7333%	25.5%	34.9285%

表 二

由信息论中信息熵的概念可知，对于具有n 个信息元素的信息集合，其熵为：

$$H = -\sum_{i=1}^{n} p_i \log p_i \qquad \cdots\cdots(12)$$

式中，p_i 表示n个状态中第 i 个状态的发生概率，H 代表消除系统不确定性所需的信息量。Shannon已证明，在没有任何干扰的条件下，一个熵为H 的信源总可以找到一种编码方法，使其编码的平均长度 L 任意接近熵H。但实际上，信息编码的平均码长L 总是大于或等于其熵值。平均码长L 为：

$$L=\sum_{i=1}^{n}P(s_i)l_i \quad \text{码符号/信源符号} \qquad \cdots\cdots(13)$$

式中$P(s_i)$为第i个汉字字串s出现的概率，l_i为字串长度。

而衡量编码是否达到最佳，则用编码效率，即，$\eta=\frac{H}{L}$，η的值越接近于1，编码效率就越高。

对于不同的*NumTop*，所形成的不同的信源，其熵值见表三。

NumTop	200	500	1000	1500	2000	3000	5000	7000
熵	9.7087	10.1408	10.6405	10.9833	11.2468	11.2468	12.1705	12.5238

表三

对于不同的信源，用Huffman编码后所得的平均码长见表四。

NumTop	200	500	1000	1500	2000	3000	5000	7000
平均码长	9.7489	10.1631	10.6766	11.0121	11.2733	11.6568	12.1981	12.5644

表四

显然，对于不同信源，其各自的编码效率均在99.6%以上。

对中文文本--23131个汉字，经过分割后，根据*NumTop*的不同，得到的字符串个数见表五。

NumTop	200	500	1000	1500	2000	3000	5000	7000
字符串数	20868	20773	20633	16501	15654	14464	12960	12189

表五

对中文文本--23131个汉字，经过分割后，根据*NumTop*的不同，得到的总体编码长度见表六。

NumTop	200	500	1000	1500	2000	3000	5000	7000
总体码长	213165	201765	197321	187080	179515	169522	157398	150719

表六

从以上所得的结果分析，对于同样的中文文本--23131个汉字，*NumTop* 不同，所得的总体编码长度也随之迥然不同。随着*NumTop* 的不断增大，Huffman 编码的平均码长也逐渐增大，按直观的想法，由此编码所得的最后总编码长度也应随*NumTop* 的增大而增大。但事实上，经过本文提出的最优匹配算法对中文文本的分割，却使总的编码长度在不断地大幅度下降。如，*NumTop* 从200增加到5000，总的编码长度减小了35.43%。从另一个角度比较，如果用等长编码方法对国家标准GB2312-80规定的6763个一、二级汉字以汉字为单位编码，则所得的二元码最短码长为$\log_2 6763 \approx 12.72$。用它对23131个汉字进行编码，用本文的算法与之相比，如对于*NumTop* 为5000，相当于本算法把23131个汉字压缩为14464个（表五），同时用平均码长为11.6568（表四）进行编码。最后等长编码方法所得总编码长度是本算法编码总长度的1.87倍。随着不断扩大训练语料的规模，并相应地增大码本容量，压缩比就能够进一步提高。

以上的实验都是对于训练文本--23131个汉字进行的。我们也把由训练文本所得的码本用于测试文本中。测试文本仍然从《中国百家新闻报刊……1994》的四月份的6.9兆字节语料中节选，字数分别为5213、8225、14097、22322。用*NumTop* 为5000码本编码，其最后总的编码长度见表七。

文本规模	5213字	8225字	14097字	22322字
总体码长	35331	55859	95947	152231

表七

所得的分割字符串个数见表八。

文本规模	5213字	8225字	14097字	22322字
字符串数	2909	4599	7901	12533

表八

应用方法所得出的*NumTop* 个权值较大的汉字字串有偶尔相互交叠缺点，这在串长度大于四时出现较多，如“会治安综”、“治安综合”、“安综合治”的相互交叠，“燃油附加”“油附加费”的相互交叠等。这种情况使得码本中各数据之间的冗余度增加了。如果能够消除这种冗余性，编码效率还能够进一步提高，总的编码长度还会更小。

本算法除了对在训练文本中出现的一部分高频汉字，根据其权值在码本中进行了编码以外（见表二中的“一字串”），对其他汉字，则均以1为权值对其编码。但汉字的出现频率也是有很大差别的。如果把汉字之间的频率差别引入编码，尤其对于测试文本，总的编码长度还会更小。

NumTop = 1000 时的权值较大汉字字串部分内容见附录一。*NumTop* = 5000 对真实文本进行分割的部分结果见附录二。

4. 结论与展望

本文提出的用于中文信息处理中的信息压缩方法是建立在信息理论基础上的．该方法通过对中文文本的模型化，把它作为离散平稳的信源．根据统计方法所得权值，形成码本。再利用离散平稳信源的组间独立、组内相关的特性，从求取汉语语料中各个词联合的最大似然度的角度，在码本的范围内对汉语语料进行分割．从实验结果可以看出，由于本方法是基于大量文本的统计基础上的，它能够细致的刻划文本中信息的分布状况。同时，又充分考虑汉字之间的相关性，因此，基于此方法所得码本建立的 Huffman 编码的编码效率极高，只要不断扩大训练语料的规模，并相应地增大码本容量，就能够进一步提高压缩比。

另外，由于本方法便于随被处理语料的增加越来越集中体现信息的分布及汉字的关联性，因此，它特别适合应用于特定领域中的高效信息传输和压缩。

在编码方面，以Huffman 码为代表的某些最优变长编码，由于在数学上缺乏构造性，难以运用现代数学理论，来解决它们编码时的工程实际问题，严重影响了这些编码方法的广泛应用。本方法为此提供了有效途径。

参考文献

[1] 刘源、谭强、沈旭昆．信息处理用现代汉语分词规范及自动分词方法，清华大学出版社、广西科学技术出版社，1994, pp. 1-56

[2] 梁南元．在论汉语自动分词和切分知识，1987，ICCIP Beijing,

[3] 黄新亚，米央，信息编码技术及其应用大全，电子工业出版社，1994, pp. 8-43.

[4] 许织新，数据压缩，国防工业出版社，1990, pp. 81-85.

附录一

的 企业 业 一 有 大 要 工作 发展 在 中 是 经济 工 会 发 企 和 作 人 社会
乡镇企业 经 了 国 市 省 市场 为 不 年 中型企业 产 地 建 国有大中 有大中型
大中型企 展 乡镇企 镇企业 政府 这 实 力 建设 全 政 公司 改革 个 电 开 行
型企业 济 动 各 宣传思想 大中型 中型企 方 上 机 社会治安 国有大 有大中
工作会 议 传思想工 思想工作 重 技术 加 定 公 们 以 出 同 社 部 委 成 乡镇
会议 改 进 我 好 市场经济 问题 镇企 场 多 来 社会主义 合 生 领导 家 面 对
工业 群众 思想 搞好国有 好国有大 业的 企业的 图书馆 工作会 宣传思 传思想 体
时 前 他们 现 党 国有 宣传 大中 综合治理 中介机构 化 到 书 思想工 想工作
生产 于 有大 理 乡 关 资 部门 我省 中型 搞好 型企 把 一个 我们 府 主 设
市场经 场经济 车 量 县 起 国家 重要 安 建材 治 高 全国 加强 者 自 调 强 报
制 区 导 燃油附加 油附加费 新 工作的 搞好国 好国有 利 技 日 思 党的 图书
机构 司 规 条 革 种 从 而 他 市政府 一项 强调 大局 要进一 党政 稳定的 各级党
依靠群众 坚 人民群众 面的 指导 党委和政 平方公里 条龙 亿元 的成 战略
一批企业 产品质量 产的 家企 轴集团工 集团工业 时期 形式

附录二

四川富益 奇 迹 的 奥 秘 本报记者 吴 中 福 王 实 编者按 年是 产权 改革 年
如今的 改革 已经 深化 到了这样 一个 层次 一 产权 的改革 如今的 也 已 扩 展
到了这样 一种 境 界 一把 产权关系 放开 在中 把产权 资源 进行 重 新 配置 和
优化 组 合 这是 改革开放 深化 个 年 头 之后 的 必 由 之路 它 表 明 只有
产权关系 按 市场经济 配置 生产要素 企业 才能 优化 才能 四川富益 电力股份
有限公司 仅仅 两年 的实践 就 给 我们 提供了 有 益 的启示 它 揭 示 出
一个企业 迈 向 新 生的 进程中 突破 单位 割据 放 弃 自己的 部 份 产权和 原
有 利益 固 然 痛 苦 但 只要 从 大处着眼 按市 经济 规 律 产权关系 便 能 换
回 大 效益 真正 解放生产 力 四川 电力 眼 下 最 紧 缺 素 有 千 水 之 省 美
誉 的四川 水资源 总 量为亿 立 方米 然 而 尽 管 经 过 多年 开发 却 仍 有
水资源 在 白白 地 流 淌 沱江 穿 流 的 富顺县 也不 例 外 据 勘 查 滔 滔 沱江
流 经 的富顺 段 可 建 个中 小 型 水电站 但 水电 投资 大 地方 投资 力不 从
心 在 富顺 地方 发电 的 记 录 上 两年前 还 挂 着 零 改革开放 风 起 云 涌
富顺县委 县 府 不 甘 心 守 着 流 银 的 江 河 再 受 穷 几年 前 他们 好
不容易 靠 借 贷 来 办 电 随 着 一 声 声 轰 轰 隆 隆 的 巨 响 一 座 投资万元
装 机 容 量 为万千瓦 的 水电站 动 工 开始 了 富顺电力 史 上 零 的突破 谁
拥有 产权和 代 销 权 这是 电力 行业 的核心 问题 以往计划 经济体制 一切 权
务 归 各级政府 部门 由此 便 有 几十 年 一 贯 制的 大小电网 割据 局 面
两年前 富顺 人 却 是 地方 政府 把 配置 权 交 给市场 由 企业 在市场 上 自 我
配置 生产要素 于是 川 电 便 有了 大小电网 协调 统一 的新 格局 希望产生 了
忧虑 也 伴 着 希望产生 早 在川 富电改组 初 县委书记 王 濯 根 就 曾 带 着 一
班 骨干 人 马 考察了 乐 山 等 地 多 座 水电站 学习 兄 弟 电站的 先进 经验
并 总结 其 面临 的 矛盾和 困 境 他们 发 现在 电站的 管理 中 大小电网 的矛盾
将 是 今 后 要 面对 的主要 矛盾 所 谓 大小电网 的矛盾 据记者了 解 就是 以

各地 电力 系统 为代表的 国 家大电网 与 各地 水电 系统 为代表的 地方电网的矛盾 矛盾 的 焦 点 是 争 夺 供 区 电 是 瞬 间 产品 卖 不 出 去 就 白白浪 费 于是 就有 了 难以 计数的电 被 低 价 收 购 又有 多 到 更 难以 计数的电 白白 流 失 这在 电 能 本 身 就 十分 紧 张 的 中国 真正 是 触 目 惊 心 的 浪费 省水电厅 分 管 地方 电力的 部门 向 记者 介 绍 他们 所 辖 的地方 小水电 共 万千瓦 地方 年发电 量为亿 千瓦时 占 省 总 发电量 的 而 地方电站 供电 地盘 则 占全 省 这些 地方电站 都 不同程度 地 与 国 家大电网 产生 着 矛盾和 磨擦 这是 不可 回 避 的事 实 邹 永 福 富顺县 政 协副主席 大小电网 的矛盾 就是 我们 常 说 的条 条 块 块 的矛盾 国家 电力部门 与 地方 水电 部门 企业 以及 地方 集 资 兴办 的 小水电 各拥有各 的产权 各拥有各 人 的利益 这些 产权 和利益 历史 遗 留 下来的 使 生产力 不能 解放 罗 旭 良 富顺县 县 长 怎 样 解决大小 电网矛盾 我们 认为 要从 根 子 上 从 入 手 通过 搞股份制 企业 把产权 部分 放开 和 转 移 使 大小电网 间 形成 产权 共同体 和利益 共同体 王 耀 根 富顺县 委书记 搞股份制 的 深 层次 问题 是 解放生产 权 力 和生产 资源 的配置 问题 谁 来 配置 以往计划 经济 靠 政府 各级政府 分 管部门 不同 利益 不同 矛盾 由此而 生 如今 我们 用 的 眼 光 看 产权 从 自身 做起 打 破 地方 割据 把 政府 的配置 功 能 放 给市场 按市场 重 新 组 合 产权关系 矛盾 就 迎 刃 而 解 了 年月 注 册 资 本 万元 的 四川富益 电力股份 有限公司 正是 在这种 形势下 诞 生了 公司 由 代表 地方电网 的国 营 富顺电力 总公司 和 两家 代表国家 电网的 自贡电 业 局电力实 业总公司 川南电 力调度 电力 技术服务 部 共同 发 起 空 前 的产权 实 体 后 形成 的 股份制 企业 其 实 质 是一种 混合经济 在这种 新的 共 生 体 中 国家 法人 股东 的产权 达到 了 前所未有 的 明 晰 三 方 的 资产 前所未有 的人 格 化 未有 地 巨大 总结 川富电改 组 一年 产生 惊 人 效益 奥 秘 时 公司 总经理 罗 崇 远 如 是 说 的混合经济 确 实 有 着 强大 的生命 力 它不仅仅 是 解决 了大小电 网的矛盾 问题 更 重要的 是 充分发挥 了 企业发展 的内在 动力 据记者了 解 全省 约 个 小电网 中 也 曾 有 过 采取 电力部门 供电 部门和 小水电 实行 联营方式 来 解决大小 电网矛盾 可 是 这种 联营方式 只 能 缓 解 而 由 的利益 矛盾 对 电力 长远发展 没有 保证 没有 后 劲 而 由 发电 供电 输 电 三 位 一 体 组成的 川富电 却 较 好 地解决 了 上 述 问题 这 在全省 是 独 一 无 二 的 创 新 模式 请 看 作为 法人股 两家 单位 一 自贡电 力 局电力实 业总公司 及 川南电 力调度 局 技术服务 部 他们 除 投资万元 股 本 外 因为 利益 一 致 风险共担 积极协 同 做好 电力 的发 输 配 工作 适 时 增加 销 量 由此而 刺 激 川富电 所 属 水电站 努力 多 发电 再 看 电站 多 职工 人 人 均 认 购 了 数量 可 观 的股份 每 个人 都 成了 企业的 股东 企业 的利益 再 也不是 他们的 身 外 之 物 而 与 他们 息 息 相 关 了 于是 一系列 根 本 性 的变化 也 就 由此 产生 一 以往 吃 惯 大 锅 饭 的企业 职工 名 义 上 是 企业 的主 人 而 实 质 上 除 按月 领 取 相 对 固 定的 工资 和 奖 金 外 很 少 关心 企业的 经营 改组后 不同 了 企业 每 个 员 工 都 拥有 企业 的股份 相 互 命 运 与 利益 紧紧 地 捆 在一起 职工 主 人 翁 精神 真正

Combining Multiword Units into a Hidden Markov Model for Part-of-Speech Tagging

Jae-Hoon Kim
Spoken Language Processing Section, ETRI,
161, Kajong-Dong, Yusong-Gu, Taejon, 305-350
KOREA
Tel: +82-42-860-6136
Fax: +82-42-861-1342
e-mail: jhoon@zenith.etri.re.kr

Abstract

The lack of lexical information involves a hidden Markov model for part-of-speech (POS) tagging in lots of difficulties in improving the performance. To alleviate the burden, this paper proposes a method for combining multiword units, which are types of lexical information, into a hidden Markov model for POS tagging. This paper also proposes a method for extracting multiword units from POS tagged corpus. In this paper, the multiword unit is defined as more than one word, which frequently makes POS tagging errors. Our experiment shows that the error reduction rate is about 13%.

1 Introduction

Part-of-speech (hereafter POS) tagging is to assign a POS to each word in a sentence. The POS tagging system is widely used in speech recognition and synthesis, information retrieval as well as natural language processing. The accuracy of most of them is at least 95%, with practically no restrictions on the input text (Church and Mercer, 1993). This means that there is a tagging error every 20 words in text tagged by the system. The tagging error can cause serious problems in several applications. In the case of the syntactic parsing system, the tagging error causes parsing errors or failure. This is the motivation of researches on improving the performance of the POS tagging by using all possible information.

A hidden Markov model (hereafter, HMM) is well-known for POS tagging. In the model, one of its problems is that it is not easy to reflect lexical contextual information (hereafter, LCI) although the LCI plays an important role in POS tagging (Kim, 1996; Lin, Chiang, and Su, 1994). For example of the word '*sound*', its POS is a noun in the sentence '*sound energy*', an adjective in the phrase '*sound fruit*', and a verb in the phrase "*They sound alarms.*" As you can see this example, some words are affected in the determination of their correct POSs by surrounding words rather than surrounding POSs. We propose a method for reflecting the LCI on an HMM to improve the performance of the tagging system. To model the LCI on the HMM, we should solve two problems: One is how to combine the LCI such as multiword units into the HMM; the other is how to determine the combined LCIs. We have slightly modified the HMM for the former problem and have introduced extraction of collocation for the latter problem. The proposed method has reduced the error rate by about 13% as compared with the original HMM. We expect that the proposed method shows the more promising result if the LCI could be made manually, but laboriously, rather than automatically like this paper.

This paper is organized as follows; In Section 2, we discuss HMM and Korean POS tagging as background works. In Section 3 and 4, combination of an HMM and multiword units as an LCI and the method for extracting the multiword units are described, respectively. After presenting some experimental results and comparing with other works in Section 5 and 6, respectively, we summarize our findings and draw conclusions in Section 7.

2 Background

2.1 HMM for POS Tagging

We introduce a probabilistic model well known as an HMM for POS tagging. In the model, a POS tagging procedure ϕ is to select the most proper POS sequence T_i satisfying with Equation (1) in a given sentence W (Allen, 1995; Charniak *etc.*, 1993; Kim, Lim, and Seo, 1995).

$$\phi(W) \equiv \text{argmax}_{T_i} \Pr(T_i|W) = \text{argmax}_{T_i} \Pr(T_i, W) \tag{1}$$

Equation (2) is derived from Equation (1) by using the Markov assumption and the chain rule, where the input sentence W is $w_1, w_2, \ldots, w_n$ and the most proper POS sequence for W is $t_1, t_2, \ldots, t_n$.

$$\phi(W) = \text{argmax}_T \prod_{i=1}^{n} \Pr(t_i|t_{i-2}, t_{i-1}) \Pr(w_i|t_i) \tag{2}$$

This equation is called the second order HMM for POS tagging. On the right side of Equation (2), the first is called a contextual probability and the second a lexical probability (Merialdo, 1994).

2.2 Korean POS Tagging

Korean is different from English in word-formation as well as word order. According to the difference, the definition of the POS tagging can vary slightly. English POS tagging assigns the most proper POS to each word in a given sentence as mentioned in Section 1 (Allen, 1995; Merialdo, 1994). On the other hand, Korean POS tagging assigns not only the most proper sequence POSs but also the most proper sequence of morphemes to each Eojeol[1] in a given sentence (Kim, Lim, and Seo, 1995)[2].

We widely use a well-known HMM for Korean POS tagging like English POS tagging. According to whether the information is included between Eojeols or not, Korean POS tagging gets divided into two models, Eojeol-based POS tagging model (Lee, 1997; Lee, 1993) and morpheme-based POS tagging model (Kim, 1996; Kim, Lim, and Seo, 1995; Lee, 1995). In the former, a tag of an Eojeol is represented as the POS sequence of morphemes (Lee, 1993) or a POS pair which is the beginning and the end of the POS sequence of morphemes (Lee, 1997) for a given Eojeol. An advantage of this model is to consider the contextual information for Eojeols as well as morphemes. On the other hand, a disadvantage is not to fix the number of Eojeol tags, therefore data sparseness and some Eojeol ambiguities on the same POS sequence arise. In the latter, the number of morpheme tags is fixed and small, but the contextual information for Eojeol can not be reflected.

Recently, to improve the performance, a hybrid model begins to appear on the stage of Korean POS tagging. As a representative example of the hybrid model, there is a model that is combined with HMM and rules like Brill's transformation (Lee and Shin, 1995; Lim, Kim, and Rim, 1997)

[1] An Eojeol is a sequence of morphemes between two spaces and is very similar to a word in English

[2] Note that readers can find the other differences in another paper of author (Kim, Lim, and Seo, 1995), but not mentioned in this paper.

Table 1: Some examples of multiword units

No.	Multiword unit (Meaning)	Remarks
1	'*hanpich cwunghakkyo*' (Hanbit Middle School)	a proper noun
2	'*cenhwa penho*' (telephone number)	a compoud noun
3	'*-ko iss-* (be -ing)	an auxiliary conjunctive ending and an auxiliary verb
4	'*-ey kwanha-*' (with regard to)	a particle and a verb
5	'*kkamccak nolla-*' (be startled all of sudden)	a adverb and a verb
6	'*kolthang mek-*' (be cheated)	a noun and a verb

3 Combination of Multiwords and HMM

In this paper, a multiword unit is defined by more than one adjacent word without regard to a grammatical unit in a sentence. Of course, most grammatical units consisting of more than one word belong to multiword units. Table 1 shows some examples of multiword units[3]. In Table 1, 1 and 2 are grammatical units, 3 and 4 a functional word and a content word, which are closely related together, and 5 and 6 some collocative words like '*take place*' in English. Except for those in Table 1, there are several sorts of multiword units as in Table 5.

In combining multiword units into an HMM, we should solve two problems: One problem is how to involve naturally multiword units in an HMM without changing the original model; The other problem is how to extract multiword units from texts or corpus. We will describe the problems in subsections in sequence.

3.1 Multiword unit based POS tagging model

A multiword-based POS tagging model, which is based on the h-order HMM, is defined by

$$\phi(W) \simeq \text{argmax}_{t_{1,n}} \prod_{i=1}^{n} \begin{cases} \Pr(t_i|t_{i-h,i-1}) \Pr(w_{i-k,i}|t_i) & \text{if } \phi = m_{i-1} \cap m_i \\ \frac{\Pr(t_i|t_{i-h,i-1})}{\Pr(t_{i-1}|t_{i-h,i-2})} \frac{\Pr(w_{i-k,i}|t_{i-k,i})}{\Pr(w_{i-p,i}|t_{i-p,i})} & \text{otherwise,} \end{cases} \quad (3)$$

where W is an input sentence, T is a correct POS sequence for W, k is the length of a multiword unit minus one, and p is the length of intersected words between a previous multiword unit and a current multiword unit. $w_{1,n}$ ($w_1 w_2 ... w_n$) denotes a sentence with n words. In a similar way, $t_{1,n}$ ($= t_1 t_2 ... t_n$) denotes a POS sequence for $w_{1,n}$. Next we want to probe relations between parameters h, k, and p of Equation (3). $k (0 \leq k < h)$ has a different value from state to state. For the original HMM, the values of k on all states are zero. Now we consider the value of k to be one. The multiword unit $w_{i-1,i}$ is denoted by m_i and consists of two words w_{i-1} and w_i. Therefore, an observation symbol on each state is a word or two words according to the value of k in case of $h = 2$. p is the length of intersected words as mentioned above and the value of p is $0 \leq p \leq k$. In Equation (3), the above equation without intersected words and the numerator of the below equation are the same with the original HMM. The denominator of the below equation, however, prevents the probabilities of the multiword units from reflecting the intersected words on the final sequence twice.

[3] In this paper, the Yale Romanization is used to represent Korean words and sentences.

Table 2: The structures of some Eojeols

Eojeols	*hakkyoey*	*kako*	*issta.*
Morphological structures	*hakkyo*/nc+*ey*/jca	*ka*/pv+*ko*/ecq	*iss*/pa+*ta*/ef+./s.
		kal/pv+*ko*/ecq	*iss*/px+*ta*/ef+./s.
		ka/px+*ko*/ecq	
		ka/pv+*ko*/ecx	
		kal/pv+*ko*/ecx	
		ka/px+*ko*/ecx	

In Figure 1 as an example, we go into details about this model with the help of Table 2, which shows the morphological structures of Eojeols in a Korean sentence "***hakkyoey kako issta***(I go to school)."[4]. The figure shows a (weighted) network (lattice) of the example sentence as an observation sequence based on the second order HMM. In the figure, a state and a transition of the HMM are represented by a node and an edge, respectively, and an observation symbol is labeled on the right and top of each node. Thus, the values on a node and an edge, but disappeared from the figure, mean a (multiword unit-based) lexical probability and a (multiword unit-based) contextual probability, respectively. The most proper sequence is represented by a bold line and \$ is a special symbol to represent the beginning and the end of a sentence in the figure.

To help readers understand this model fully, we explain this model in detail through a concrete example. Suppose that h be 2 for convenience. So, k can be 0 or 1. For a given Korean sentence "***hakkyoey kako issta.***" which is the same sentence given in the example in Figure 1, the valid morphological analysis is "*hakky*/nc+*ey*/jca *ka*/pv+*ko*/ecx *iss*/px+*ta*/ef+./s." which is underlined in Table 2. Suppose that '*-ko iss-*' be a multiword unit with $k = 1$. All words except this word are simple words (morphemes) with $k = 0$. Then, the probability $\Pr(hakkyo + ey \;\; ka + ko \;\; iss + ta + .,$ nc+jca pv+ecx px+ef+s.) of Equation (3) is calculated by the followings;

$$\begin{aligned}
&\Pr(hakkyo|\text{nc})\,\Pr(\text{nc}|\$, \$)\times \\
&\Pr(ey|\text{jca})\,\Pr(\text{jca}|\$, \text{nc})\times \\
&\Pr(ka|\text{pv})\,\Pr(\text{pv}|\text{nc, jca})\times \\
&\Pr(ko|\text{ecx})\,\Pr(\text{ecx}|\text{jca, pv})\times \\
&\frac{\Pr(ko,iss|\text{ecx, px})\,\Pr(\text{ecx, px}|\text{pv})}{\Pr(ko|\text{ecx})\,\Pr(\text{ecx}|\text{pv})}\times \\
&\Pr(ta|\text{ef})\,\Pr(\text{ef}|\text{ecx, px})\times \\
&\Pr(.|\text{s.})\,\Pr(\text{s.}|\text{pv, ef})\times \\
&\Pr(\$|\$)\,\Pr(\$|\text{ef, s.})
\end{aligned} \tag{4}$$

3.2 Parameter estimation of multiword units based POS tagging model

If k is 0, parameter estimation is the same with the original HMM. Now we turn to parameter estimation in case of $k = 1$. Consider a special case as an example in case of $k = 1$ and $h = 2$ for our experiment described below. Lexical probability and contextual probability for a multiword unit $w_{i-1,i}$ are estimated by Equation (5) and (6);

$$\Pr(w_{i-1,i}|t_{i-1,i}) \simeq \frac{C(w_{i-1,i}, t_{i-1,i})}{C(t_{i-1,i})} \tag{5}$$

$$\begin{aligned}
&\Pr(t_{i-1,i}|t_{i-2}) \simeq \tfrac{C(t_{i-2,i})}{C(t_{i-2})} \quad \text{if } k = 1 \\
&\Pr(t_i|t_{i-2,i-1}) \simeq \tfrac{C(t_{i-2,i})}{C(t_{i-2,i-1})} \quad \text{otherwise}(k = 0),
\end{aligned} \tag{6}$$

[4]A morphological structure is a result of morphological analysis for an Eojeol and is regarded as a linear structure of which elements are distinguished by a delimiter '+'. '*hakkyo*/nc' means that the POS tag of the morpheme '*hakkyo*' are 'nc'. We put the list of Korean POS tags in an appendix.

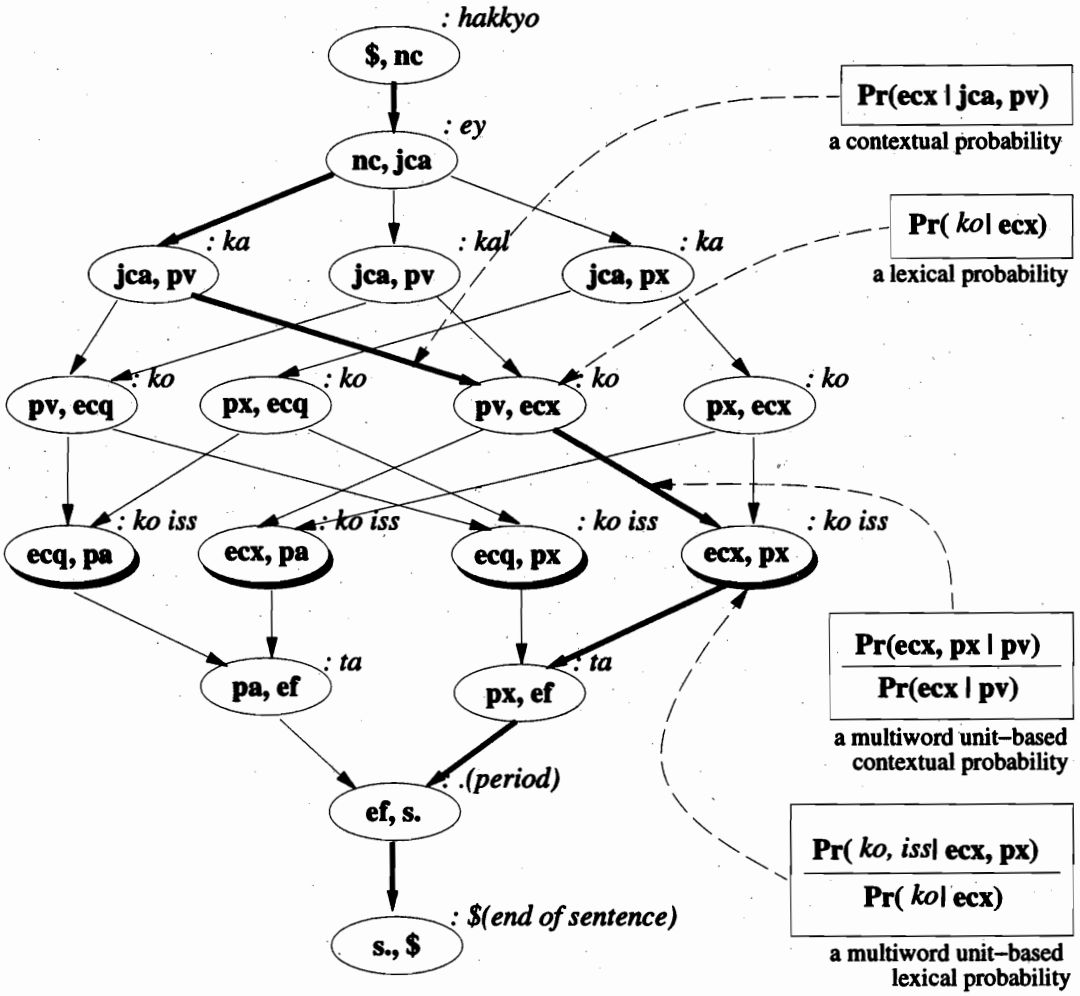


Figure 1: A weighted network (lattice) of observations and states (nodes) based on the second order HMM

where $C(x)$ is a frequency of x. 3-gram for POSs is sufficient to estimate parameters for contextual probability in the second order HMM and $k = 1$ as you can see in Equation (6). Therefore, the degree of the data sparseness is the same with the original HMM. In proportion to using some kinds of multiword units, however, the lexical probability is very different from the contextual probability in data sparseness, which might cause the performance to make worse. To alleviate this problem, we use multiword units including an error-prone word with high frequency. We will describe the extraction method of multiword units in the next section in detail.

3.3 Extraction of multiword units

The extraction method of multiword units is similar to that of collocation in respect of using the frequency of n-gram (Lee, Kim, and Kim, 1995). This is the difference in that the frequency is counted in not all words, but only in error-prone words. That is, if w_i is an error-prone word, the frequency of multiword units including w_i is defined by Equation (7);

$$C_e(w_{i-k,i}) \text{ or } C_e(w_{i,i+k}) > \rho_1, \tag{7}$$

where w_i is an error-prone word, $C_e(w_{i-k,i})$ and $C_e(w_{i,i+k})$ are the frequency of the left context and the right context of the error-prone word w_i, respectively, and $\rho_1(\rho_1 > 1)$ is a constant as a threshold. Generally, mutual information is used for extracting collocation (Church and Hanks, 1990), but is improper in extracting multiword units. This is the reason that the left and the right context are considered differently. Let us consider the left context of an error-prone Korean word '*-ko*'. The left context can be all verbs such as '*mek* (eat) + *ko*' and '*ip* (wear) + *ko*' *etc.* In this paper, these verbs are improper as multiword units based on '*-ko*'. On the other hand, consider the right context of the word '*-ko*'. In many cases, the right context is a special auxiliary verb 'iss-', but this might not always be the case. Therefore, in the case of the error-prone word '-ko', the left context is not proper as a multiword unit, but the right context is proper. In this paper, as we pay attention to this point, the conditional probability and the relative frequency count (Su, Wu, and Chang, 1994) are used for extracting multiword units as in Equation (8).

$$\Pr(w_{i-k,i-1}|w_i)\frac{C(w_{i-k,i})}{\mathrm{E}\{C(w_{i-k,i})\}} \text{ or } \Pr(w_{i+1,i+k}|w_i)\frac{C(w_{i,i+k})}{\mathrm{E}\{C(w_{i,i+k})\}} > \rho_2, \tag{8}$$

where $\mathrm{E}\{C(w_{i-k,i})\}$ and $\mathrm{E}\{C(w_{i,i+k})\}$ are the average frequency of $w_{i-k,i}$ and $w_{i,i+k}$, respectively, $\rho_2(\rho_2 > 0)$ is a constant as a threshold, w_i is an error-prone word. In this paper, ρ_1 and ρ_2 are controlled to keep minimal errors on the training corpus described in next section in detail.

4 Experiment and Evaluation

The main objective of this paper is to show that the multiword unit is a kind of useful information to improve the performance of a tagging system, especially based on an HMM.

4.1 Experimental Environment

We use the "KAIST corpus" data described in Kim (1996). It contains 15,950 sentences and its other statistics are shown in Table 3. These sentences have been tagged manually at the department of computer science in KAIST. The training corpus and the test corpus are independent. We use 51 different POS tags as in Appendix. We have built a dictionary that indicates the list of possible tags for each morpheme, by taking all the words that occur in the total

Table 3: Statistics of training and test corpus

statistics	training corpus	test corpus
no. of sentences	12,082	3,868
no. of Eojeols	131,581	41,122
no. of morphemes	284,241	88,683
avg. no. of Eojeols per sentence	10.89	10.63
avg. no. of morphemes per Eojeol	2.16	2.16

Table 4: Performance according to model parameters

no. multiword units	ρ_1	ρ_2	no. of errors		for morphemes
			Eojeol	morphemes	error reduction rate(%)
0	–	–	1655	1987	0.00
29	5	3.000	1606	1889	4.93
43	5	1.000	1601	1885	5.13
60	3	1.000	1589	1876	5.59
78	2	1.000	1591	1878	5.49
120	3	0.100	1533	1786	10.12
130	3	0.050	1507	1749	11.98
143	**3**	**0.010**	**1493**	**1733**	**12.78**
146	3	0.005	1493	1733	12.78
151	3	0.001	1495	1737	12.58

corpus. In similar way, we have established a multiword unit dictionary by using the extraction method described in Section 3.3. Thus, these are a closed dictionary since a word will not have all its possible tags although the tags actually are within the corpus. In Korean, a morphological analyzer plays an important role in POS tagging. We used the morphological analyzer based on lexicalized morphotactics (Kim, 1996) for our experiment.

4.2 Performance evaluation

In this experiment, we extracted 3-gram of POS from the training corpus. Then, we computed the relative frequency count as the supervised parameter estimation method and used the Good-turning method (Good, 1953) for smoothing. This model was then used to tag the test sentence in the test corpus. The results are indicated in Table 4. The table shows that the performance varies as the control of two model parameters, ρ_1 and ρ_2. Note that the first row on the table is the performance concerned in the second order original HMM. In our experiment, the number of selected multiword units is determined according to the value of ρ_1 and ρ_2 in the training corpus. We get the best result in the case of $\rho_1 = 3$ and $\rho_2 = 0.01$. As a result, the error reduction rate is about 13%. Total tagging accuracy is about 98%, and a gain of 0.2% in accuracy is produced.

4.3 Selected multiword units

In our experiment, Table 5 shows a part of the selected multiword units of which some are not intuitive. In the table, the functional words are marked with an asterisk '*'. A selected multiword unit has at least one functional word. This means that most error-prone words are functional words in Korean. A great number of endings are especially error-prone functional words. The determination of correct POS for the endings requires syntactic analysis, but it is

Table 5: A part of the extracted multiword units in our experiment

Left context of error-prone word		Right context of error-prone word	
$(w_{i-1}$	$w_i)$	$(w_i$	$w_{i+1})$
*i**	*ta**	*ha*	*n**
ha	*e**	*ke**	*i**
*ass**	*ta**	*ha*	*e**
*i**	*nka**	*key**	*twi*
ha	*ko**	*ha*	*ess**
ha	*key**	*ha*	*ko**
*key**	*twi*	*ko**	,
*nun**	*ke**	*ha*	*nun**
*e**	*poi*	*il*	*i**
*e**	*naka*	*sulep**	*n**
*i**	*ya**	*ha*	*l**
ha	*mye**	*twi*	*ess**
*i**	*lan**	*ha*	*nta**
*n**	*il*	*twi*	*e**
*ey**	*ilu*	*m**	*ul**
*ul**	*tut*	*yeph*	*ey**
*lul**	*tut*	*key**	*ha*

somewhat, but not completely, resolved by observing some words around the error-prone endings. A representative example is a phrase constituted by an auxiliary conjunctive ending and an auxiliary verb.

5 Discussions

For POS tagging, a VMM (variable Memory Markov) model proposed by Schütze and Singer (1994) is similar in using variable-length context to our method. Both methods also adjust the length of context using errors. In order to determine the context, Schütze and Singer use the statistical error based on relative entropy, while we use the error environment including at least one error-prone word based on the conditional probability and relative frequency count. Another difference is a type of variable contexts, that is, they use only POSs while we use LCIs as well as POSs. Brill's method(Brill, 1995) can also accept variable contexts. It, furthermore, have the nature of long-distance correlations as well, but our proposed methods neglect it due to the Markov nature. This is a drawback of our proposed methods. There is another tagging model with variable context, which is called PCM (probabilistic classification model) proposed by Lin, Chiang, and Su (1994). PCM is also similar to our proposed method in applying to error-prone words. PCM re-tags POSs to error prone words selected by CART while our method do not.

Now we turn to a method for extraxting multiword units, which is very similar to that for extracting collocations(Church and Hanks, 1990; Smadja, 1993; Su, Wu, and Chang, 1994). Especially our approach is similar in using relative frequency count to the approach proposed by Su, Wu, and Chang (1994). We, however, use the conditional probability as mentioned in Section 3.3. We observe that the conditional probability is good for extracting the selectional restrictions through another experiment(Lee, Kim, and Kim, 1995).

6 Conclusion Remarks

In this paper, we have presented a POS tagging model with combining multiword units into an HMM and a method for extracting multiword units from POS tagged corpus. In this paper, the multiword units are defined as more than one word which frequently cause POS tagging errors.

Our experiment shows an error reduction rate of about 13% as compared with the original HMM and a total accuracy of about 98%. The results of experiments reveal that multiword units are well-suited to a type of the lexical contextual information on an HMM. We expect that the proposed method shows the more promising results if multiword units (not selected automatically, but error-prone words explicitly) could be added manually, but laboriously, rather than automatically.

Acknowledge

The author acknowledges valuable advice with the anonymous reviews. This work has been supported by the Ministry of Information and Communication under the title of "Spoken Language Translation under Multimedia Environment."

References

Allen, J. *Natural Language Understanding*, 2nd edition, The Benjamin/Cummings Publishing Company, Inc.

Brill, E. (1995), "Transformation-based error driven learning and natural language processing: a case study in part-of-speech tagging," *Computational Linguistics*, vol. 21, no. 4, pp. 543-564.

Charniak, E., Hendrickson, C., Jacobson, N., and Perkowitz, M. (1993) "Equations for part-of-speech tagging," *Proceedings of National Conference on Artificial Intelligence(AAAI-93)*, pp. 748-789.

Church, K. W. and Hanks, P. (1990), "Word association norms, mutual information, and lexicography," *Computational Linguistics*, vol. 16, no. 1, pp. 22-29.

Church, K. W. and Mercer, R. L. (1993), "Introduction to the special issue on computational linguistics using large corpora," *Computational Linguistics*, vol. 19, no. 1, pp. 1–24.

Good, I. J. (1953), "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, no. 3, pp. 237-264.

Kim, J.-D., Lim, H.-S., and Rim, H.-C. (1996), "A Korean part-of-speech tagging model based on morpheme unit with Eojeol-unit context," *Proceedings of the Korea Cognitive Science Society Spring Conference*, pp. 97-106(in Korean).

Kim, J.-H. (1996), *Lexical Disambiguation Using Error-Driven Learning*, Ph.D. Thesis, Dept. of Computer Science, KAIST(in Korean).

Kim, J.-H., Lim, C.-S., and Seo, J. (1995), "An efficient Korean part-of-speech tagging using hidden Markov model," *Journal of the Korea Information Science Society*, vol. 22, no. 1, pp. 136-146(in Korean).

Lee, H.-K. (1997), "An effective estimation method for lexical probabilities in Korean lexical disambiguation," *Journal of the Korea Information Processing Society*, vol. 3, no. 6, pp. 1588-1597(in Korean).

Lee, S.-H. (1995), *A Korean Part-of-Speech Tagging System with Handling Unknown Words*, M.S. Thesis, Dept. of Computer Science, KAIST(in Korean).

Lee, W.-J. (1993), *Design and Implementation of an Automatic Tagging System for Korean Texts*, M.S. Thesis, Dept. of Computer Science, KAIST(in Korean).

Lee, K. J., Kim, J.-H., and Kim, G. C. (1995), "Extracting collocations from tagged corpus in Korean," *Proceedings of the 22nd KISS Spring Conference*, Inha Univ., Incheon, pp. 623-626(in Korean).

Lee, J.-H. and Shin, S.-H (1995), "TAKTAG: Two phase learning method for hybrid statistical/rule-based part-of-speech disambiguation," *Proceedings of the 6th International Conference on Computer Processing of Oriental Languages(ICCPOL-95)*, Hawaii.

Lim, H.-S., Kim, J.-D, and Rim, H.-C. (1997), "A Korean part-of-speech tagger using transformation-based error-driven learning," *Proceedings of the 7th International Conference on Computer Processing of Oriental Languages(ICCPOL-97)*, Hong Kong Baptist Univ. pp. 456-459.

Lin, Y.-C., Chiang, T.-H., and Su, K.-Y. (1994), "Automatic model refinement - with an application to tagging," *Proceedings of International Conference on Computational Linguistics(COLING-94)*, Kyoto, Japan, pp. 148-153.

Merialdo, B. (1994), "Tagging English text with a probabilistic model," *Computational Linguistics*, vol. 20, no. 2, pp. 156–171.

Shin, J.-H., Han, Y. S., Park, Y. C., and Choi, K.-S., (1994) "A HMM part-of-speech tagger for Korean wirh word phrasal relations" *Proceedings of International Conference on Recent Advances in Natural Language Processing(RANLP-94)*, Sophia, Bulgaria.

Schütze H. and Y. Singer (1994), "Part-of-speech tagging using a variable memory Markov model," *Proceedings of the 26th Annual Meeting of the Assoc. for Computational Linguistics(ACL-32)*, pp. 181-187.

Smadja, F. (1993), "Retrieving Collocations from Text: Xtract," *Computational Linguistics*, vol. 19, no. 1, pp. 143-177.

Su, K.-Y., Wu, M.-W., and Chang, J.-S. (1994), "A corpus-based approach to automatic compound extraction," *Proceedings of the 32th Annual Meeting of the Assoc. for Computational Linguistics(ACL-94)*, pp. 242-247.

Appendix: Korean POS tags

1	s,	comma	2	s.	sentence closer
3	s‘	left quotation and parenthesis mark	4	s’	right quatation and parenthesis mark
5	s-	connection mark	6	su	unit
7	sw	currency	8	sy	other symbols
9	f	foreign word	10	nca	active common noun
11	ncs	stative common noun	12	nc	common noun
13	nq	proper noun	14	nbu	unit bound noun
15	nb	bound noun	16	npp	personal pronoun
17	npd	demostrative pronoun	18	nnn	number
19	nn	numeral	20	pv	verb
21	pad	demonstrative adjective	22	pa	adjective
23	px	auxiliary verb	24	md	demonstrative adnoun
25	mn	numeral adnoun	26	m	adnoun
27	ad	demonstrative adverb	28	ajw	word-conjunctive adverb
29	ajs	sentence-conjunctive adverb	30	a	adverb
31	i	interjection	32	jc	case particle
33	jcm	adnominal case particle	34	jcv	vocative case particle
35	jca	adverbial case partcle	36	jcp	predicative case particle
37	jx	auxiliary particle	38	jj	conjunctive particle
39	ecq	equal conjunctive ending	40	ecs	subordinative conjunctive ending
41	ecx	auxiliary conjunctive ending	42	exm	adnominal ending
43	exn	nominal ending	44	exa	adverbial ending
45	efp	prefinal ending	46	ef	final ending
47	xn	noun suffix	48	xpv	verb-derived suffix
49	xpa	adjective-derived suffix	50	xa	adverb-derived suffix
51	sp	a space, specical tag			

A Robust Keyword Spotting System for Mandarin Speech

Chung-Hung Chien and Hsiao-Chuan Wang

Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan, 30043

Abstract

This paper introduces a method for designing a robust Mandarin keyword spotting system. Keywords which will be extracted from an uttered sentence are modeled by sequences of states. These state models that represent the subsyllables of Mandarin speech are generated by using the existing speech database. The non-keyword portions of an input utterance are filtered out by filler models. A simplified signal bias removal technique is applied to overcome the influences due to channel distortion and speaker variation. State integrated Wiener filters are used for noise compensation. Proposed techniques are evaluated by several experiments to show their effectiveness for robust speech recognition.

1 Introduction

In many applications, an input utterance can be recognized by extracting its keywords without transcribing all the sentence. This keyword spotting technique allows a speaker to talk to a machine naturally. Without complicated recognition algorithm, such as the continuous speech recognition, the keyword spotting method provides an alternate implementation of speech input. For small vocabulary applications, keyword spotting is an effective method for implementing a voice input system. Many researchers have been attracted into this area and

---

This research has been sponsored by the National Science Council, ROC, under contract number NSC-862745-E-007-010.

developed some remarkable keyword spotting methods [Wilpon, Miller, and Modi 1991, Wilcox and Bush 1992, James and Young 1994, Huang, Wang, and Soong 1994, Sukkar and Lee 1996] .

A typical keyword spotting method is based on hidden Markov model (HMM) technique [Wilpon, Rabiner, Lee, and Goldman 1990] where a word or a subword is modeled as a Markov chain of states. The filler models are used for filtering out the non-keyword portion of the utterance. Usually, the filler models can be generated by using speech data of selected non-keywords. A decision making scheme must be provided for discriminating those non-keyword speech and silence in an utterance. The performance of a keyword spotting algorithm depends on the effectiveness of screening the non-keyword portion.

In this paper, a keyword spotting method for Mandarin speech is introduced. The continuous density HMM technique is applied. An utterance is modeled by a finite state network composed of keyword models and filler models. Viterbi decoding algorithm is used to find the optimal state sequence and then the score of the utterance is calculated. A likelihood ratio test is applied to extract the keyword from an input utterance. Both keyword models and filler models are generated by using the state models trained by an existing speech database [Hwang, Cheng, and Wang 1996]. This implies that a keyword spotting system can be implemented by using the existing state models so that no training procedure is necessary in building an application system.

In order to overcome the channel distortion and the speaker variation, a simplified signal bias removal (SBR) algorithm [Rahim and Juang 1996] is developed. The background noise is compensated by using state integrated Wiener filters [Vaseghi and Milner 1997]. This robust keyword spotting system has been implemented on a personal computer to demonstrate its capability in voice response applications.

2 Mandarin Syllables and their Hidden Markov Models

Mandarin speech is a tonal and syllabic language. Each Chinese character corresponds to a syllable. There are about 1300 distinctive syllables in Mandarin speech. Without concerning the tones, the number of base syllables is 408. Usually we represent a Mandarin syllable as an Initial-Final model. The *final* portion is its rhyme part, and the *initial* portion is a consonant. Some of syllables are vowel only, and no consonant appears in the *initial* part. We refer these syllables as *null-initials*. Since the acoustic characteristic of *the initial* is affected by its following *final,* we consider the *initial* a context-dependent unit. Totally, there are 38 context-independent *finals* and 99 right-context-dependent *initials* in Mandarin speech. Also, there are 33 syllables of *null-initials.* In this study, the *initials* are modeled by 3-state HMMs, and the *finals* by 4-state HMMs. For a syllable of *null-initial,* its *initial* part id modeled by a 2-state HMM. Including a silence state, there are totally 498 states must be modeled.

The speech database for generating the state models consists of 5045 phonetically balanced Mandarin words spoken by 51 males and 50 females. It includes 408 base syllables in Mandarin speech. These speech data were recorded in an office room via a high-quality microphone. The speech signal was sampled at 8 kHz with 16 bits per sample. The speech signal is pre-emphasized and then a Hamming window of 256 sampling points is applied before calculating its cepstral coefficients. The frames are spaced by 128 sampling points. For each frame, a 12-order cepstrum is extracted. A feature vector consists of 12 cepstral coefficients, 12 delta cepstral coefficients, a delta log-energy, and a delta delta log-energy. Finally, an utterance is represented by a sequence of 26-dimensional feature vectors. The state model is a mixture of Gaussian densities. 498 state models are generated using the speech database described above.

3. Scoring Method for Keyword Spotting

In this study, we assume that an input utterance contains one keyword only. Then an utterance is modeled by a network structure such that a sequence of nodes representing a keyword is preceded by a filler model and followed by another filler model. The silence state is added to the beginning and ending nodes of this network for filtering out the silence portions before and after the speech. The filler model is for filtering out the garbage speech in the utterance.

For an input utterance, Viterbi decoding is applied to calculate the maximum likelihood score and to obtain its corresponding optimal state sequence. Along the optimal state sequence, the local likelihood scores belonging to the keyword states are accumulated as a keyword score.

$$L(O^v, S_k^v) = \log P(O^v | S_k^v) = \sum_{i=j}^{j+M_v-1} \log P(o_i^v | s_{k,i}^v), \qquad (1)$$

where $O^v = \{o_j^v, o_{j+1}^v, \cdots o_{j+M_v-1}\}$ are the feature vectors of the frames belonging to keyword v, $S_k^v = \{s_{k,j}^v, s_{k,j+1}^v, \cdots s_{k,j+M_v-1}^v\}$ is the corresponding states belonging to keyword v, M_v is the number of frames belonging to keyword v, and j is the starting frame of the keyword. If the likelihood scores of those decoded keyword states are calculated based on filler models, we obtain a normalization score.

$$L(O^v, S_f) = \log P(O^v | S_f) = \sum_{i=j}^{j+M_v-1} \max_{s_i \in \{S_f\}} \log P(o_i^v | s_i), \qquad (2)$$

where S_f is a set of the filler states. Then we define a likelihood ratio as follows,

$$L(O^v) = (\log P(O^v | S_k^v) - \log P(O^v | S_f)) / M_v. \qquad (3)$$

This ratio will be used for determining the recognized keyword,

$$v^* = \max_v \{L(O^v)\}. \qquad (4)$$

In order to screen out those cases of abnormal keyword duration, we set a limit to bound the keyword duration in a reasonable range. When a keyword detected in an utterance is out of the bound, this keyword is wrong and the utterance is indicated as no keyword existing.

4. Compensation of Channel Effect

The channel effect may be due to a telephone line or a microphone. We can consider the channel effect a convolution noise. In frequency domain, the resulted speech signal is expressed as

$$Y(\omega) = H(\omega)X(\omega) \ , \tag{5}$$

where $Y(\omega)$ is the distorted speech, $H(\omega)$ is the channel effect, and $X(\omega)$ is the original speech. When Eq.(5) is transformed into cepstral domain, the channel effect becomes an additive term,

$$c_y(n) = c_x(n) + \delta(n) . \tag{6}$$

When an utterance is represented by a sequence of feature vectors, and the feature vector consists of cepstral coefficients and delta cepstral coefficients, the bias can be assumed to be an additive constant vector.

$$c_{y,t} = c_{x,t} + b \ , \tag{7}$$

where $c_{y,t}$ is the feature vector of distorted speech in t-th frame, $c_{x,t}$ is the feature vector of original speech in t-th frame, and b is a bias vector. The procedure for finding the bias vector is as follows [Rahim and Juang 1996];

(a) Apply Viterbi decoding on the test utterance to find its optimal state sequence, $S = \{s_1, s_2, \cdots s_T\}$.

(b) Apply following equation to estimate the bias vector,

$$b = \frac{1}{T}\sum_{t=1}^{T}(c_{y,t} - m_i) , \tag{8}$$

where m_i is the mean of state model i corresponding to the decoded state s_t in t-th frame, and T is the number of frames in the utterance.

When the bias vector is obtained, we can apply this bias to adapt all the state models,

$$\tilde{m}_i = m_i + b \ . \tag{9}$$

Viterbi decoding algorithm is applied again to the test utterance based on adapted models. This procedure, i.e. Eq.(8) and Eq.(9), can be iterated so that a converged bias vector is obtained and all the state models are adapted to new ones. Finally, the input utterance is recognized based on the new state models. Usually, two iterations is enough to obtain converged bias vector. If training utterances are used for finding this constant bias vector, the adaptation operation is not necessary during the recognition phase. Speaker adaptation is exactly similar to channel compensation with given training utterances by a specific speaker.

5. Compensation of Additive Noise

The additive noise can be modeled as adding a noise term to the clean speech in frequency domain.

$$Y(\omega) = X(\omega) + N(\omega), \tag{10}$$

where $X(\omega)$ is the clean speech, and $N(\omega)$ is the additive noise. Many noise compensation methods have been developed for compensating the additive noise. Here we use Wiener filter to minimize the effect of noise [Vaseghi and Milner 1997]. In frequency domain, Wiener filter is expressed as

$$W(\omega) = \frac{P_{xx}(\omega)}{P_{xx}(\omega) + P_{nn}(\omega)}, \tag{11}$$

where $P_{xx}(\omega)$ and $P_{nn}(\omega)$ are the power spectrum densities of original speech and noise, respectively. When a noisy speech is input to Wiener filter, the output would be

$$\tilde{X}(\omega) = W(\omega) Y(\omega). \tag{12}$$

In cepstral domain, Eq.(12) becomes

$$c_{\tilde{x}}(n) = c_w(n) + c_y(n), \tag{13}$$

where

$$c_w(n) = c_{P_{xx}}(n) - c_{P_{xx}+P_{nn}}(n) \tag{14}$$

In our speech recognition system, Wiener filter is estimated and applied to state models to adapt the models to noisy environment,

$$\tilde{m}_i = c_{P_{m_i}} - c_w, \tag{15}$$

where

$$c_w = c_{P_{m_i}} - c_{P_{m_i}+P_{nn}}. \tag{16}$$

P_{m_i} is the power spectrum density calculated for each state model during the training phase. Its corresponding cepstrum is $c_{P_{m_i}}$. P_{nn} is the power spectrum density of noise which is estimated under silence input. Once P_{nn} is obtained and P_{m_i} is available, $c_{P_{m_i}+P_{nn}}$ can be calculated. In our implementation, P_{nn} is calculated once in a stationary noisy environment.

During recognition phase, a signal-to-noise ratio (SNR) is calculated for each utterance to adjust P_{nn}.

6. Experiments

Some experiments were conducted to demonstrate the keyword spotting algorithm and the effectiveness of our channel and noise compensation methods.

Experiment 1

Twenty city names in Taiwan were designated as keywords embedded in the uttered sentences. Six speakers each provided 50 test utterances through microphones. Only one keyword was embedded in each utterance. The garbage speech might appear before and after the keyword. The accuracy was 94.7% for mixture number is 2 for each state model.

Experiment 2

Twenty city names in Taiwan were designated as keywords embedded in the uttered sentences. Thirteen speakers each provided 20 test utterances through telephone system. Only one keyword embedded in each utterance. The garbage speech might appear before and after the keyword. The mixture number was 2 for each state model. The accuracy was 57.9% without channel compensation. The accuracy increased to 82.6% when the proposed channel compensation method was applied. The improvement was 24.7%.

Experiment 3

Thirty city names in Taiwan were designated as keywords embedded in the uttered sentences. Fifteen speakers each provided 20 test utterances through microphones. Only one keyword embedded in each utterance. The garbage speech might appear before and after the keyword. The mixture number was 2 for each state model. In order to simulate various noise conditions, test utterances were added by different noises with specific SNRs. The types of noises included white noise, factory noise, car noise and babble noise. The accuracy for various SNRs was summarized in the following table.

Table: Recognition Accuracy (%)

noise type \ SNR	0dB		10dB		20dB	
	no compe	compens	no compe	compens	no compe	compens
white noise	8.57	13.2	37.0	45.4	68.2	77.5
factory noise	16.1	24.3	55.7	58.6	79.6	82.6
car noise	65.4	78.2	77.1	81.1	81.8	82.5
babble noise	10.7	19.6	48.9	59.3	73.9	78.9
AVERAGE	25.2	33.8	54.7	61.1	75.9	80.4

The effectiveness of noise compensation depends on the type of additive noise. The result shows that car noise does not influence to much on the recognition accuracy. White noise is the most serious one because it affects a wide band of signal spectrum. In average the proposed noise compensation method can gain an improvement of 4.5% to 8.6%.

7. Conclusion

A robust Mandarin keyword spotting system is presented. Keyword and filler models can be generated by using the existing speech database. This allows user to define their own application systems. A simplified signal bias removal technique is applied to overcome the influences due to channel distortion and speaker variation. State integrated Wiener filters are used for noise compensation. Experiments show that the channel compensation can gain an accuracy improvement of about 25%, and the noise compensation can improve 8.6% accuracy in average when the SNR is 0dB.

References

Huang, E.F., H.C. Wang, and F.K. Soong, "A fast algorithm for large vocabulary keyword

spotting application," IEEE Trans. SAP, vol. 2, no. 3, pp. 449-452, 1994.

Hwang, T.H., H.M. Cheng, and H.C. Wang, "Keyword spotting for Mandarin speech based on subsyllable models," Int. Conf. Multiple Information processing, Hsinchu, Taiwan, 1996.

James, D.A. and S.J. Young, "A fast lattice-based approach to vocabulary independent word spotting," ICASSP-94, Adelaide, Australia, 1994.

Rahim, M.G. and B.H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition," IEEE Trans. SAP, vol. 4, no. 1, pp. 19-30, 1996.

Sukkar, R.A. and C.H. Lee, "Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition," IEEE Trans. SAP, vol. 4, no.6, pp. 420-429, 1996.

Vaseghi, S.V. and B.P. Milner, "Noise compensation methods for hidden Markov model speech recognition in adverse environments," IEEE Trans. SAP, vol. 5, no. 1, pp. 11-21, 1997.

Wilcox, L.D. and M.A. Bush, "Trainig and search algorithm for interactive wordspotting system," ICASSP-92, San Francisco, CA, 1992.

Wilpon, J.G., L.G. Miller, and P. Modi, "Improvements and applications for keyword recognition using hidden Markov models," ICASSP-91, Toronto, Canada, 1991.

A First Study on Mandarin Prosodic State Detection

Yuan-Fu. Liao, Wern-Jun Wang, Shu-Ling Lee, Sin-Horng Chen
Institute of Communication Engineering
National Chiao Tung University, HsinChu, Taiwan, ROC
TEL: +886-3-5711431, FAX: +886-3-5710116, Email: schen@cc.nctu.edu.tw

Abstract

In this paper, a method to detect prosodic phrase structure of Mandarin speech is proposed. It first employs an RNN to discriminate each input frame of an utterance among three broad classes of syllable initial, syllable final, and silence. Outputs of the RNN are then used to drive an FSM for segmenting the input utterance into four types of segment. They include three stable-segment - I (initial), F (final), and S (silence), and a transition-segment - T (transition). Appropriate modeling features are thus extracted from the vicinities of F-segments, and used to model the prosodic states for inter-F-segment intervals. Two prosodic-state modeling schemes are studied. One uses VQ to encode the modeling features and directly classify inter-F-segment intervals into 8 prosodic states. The other uses an RNN, trained with relevant linguistic features as output targets, to implicitly represent the prosodic status by the outputs of its hidden layer. Prosodic states can be obtained by vector-quantizing the outputs of the hidden layer of the RNN. Experimental results showed that linguistically meaningful interpretations of these prosodic states can be observed.

1. Introduction

Continuous speech contains the actual words spoken as well as supra-segmental information, such as stress, timing structure, and fundamental frequency (F0) contour patterns. This information is generally referred to as the prosody of the speech, which is affected in turn by the sentence type, the syntactical structure, the semantics, the emotional state of the speaker,

etc [Sagisaka 1996]. Traditional speech recognition methods have totally neglected this prosodic information in their recognition process. However, prosodic modeling gets more and more attentions in recent years in the area of speech recognition. Many researches have been devoted to the exploration of relevant prosodic cues from input speech utterance with the purpose of assisting in speech recognition [Compbell 1993, Kompe 1995, Wang 1994, Wightman 1994]. A prosodic model can be generally defined as a mechanism to describe the relationship between the acoustic features extracted from the prosodic parameter contours representing the prosodic phrase structure of speech and the linguistic features extracted from the associated text. Two basic types of prosodic model can be found. One is a model designed to detect the prosodic phrase structure of an utterance by using some features extracted from the prosodic parameter contours. Its main purpose is to provide an additional score to help speech recognition. The other type of prosodic model is designed to predict the embedded prosodic phrase structure from a text by using some linguistic features extracted from the text. Obviously, it is mainly used in text-to-speech to help the generation of prosodic information for synthesizing natural speech [Chen 1996a].

In this paper, we are interested in the first type of prosodic model. A method to detect the prosodic phrase structure of the input speech is proposed. Our final goal is to derive a prosodic model in the pre-processing stage of a speech recognizer for the use in the following recognition process to assist in speech recognition. In this preliminary study, only the prosody modeling is discussed. The primary problem encountered in this study is how to extract appropriate modeling features from the input speech under the constraint that *a priori* information about syllable or word boundaries is not available [Wightman 1994]. The problem is solved by first dividing the input utterance into labeled-segments, and then extract modeling features from stationary voiced segments.

The organization of the paper is stated as follows. Section 1 briefly describes background information and states the problem. Section 2 presents the proposed method of prosodic-state detection. Experimental results are given in Section 3. Conclusions are given in the last section.

2. The proposed method

Fig. 1 shows a block diagram of the proposed method of prosodic-state detection. It consists of five main parts: spectral feature extraction, recurrent neural network (RNN) pre-classifier, finite state machine (FSM) based segmentation [Chen 1996b], modeling feature extraction, and prosodic state classification. Input speech signal is first divided into frames. Some spectral features are then extracted for each frame. An RNN is then employed to discriminate each input frame among three broad classes. They include syllable initial, syllable final, and silence. Outputs of the RNN are then used to drive an FSM to segment the input utterance into four types of segment. They include three stable-segment types of I (initial), F (final), and S (silence), and a transition-segment type of T (transition). Appropriate modeling features are then extracted from the vicinities of F-segments, and used to model the prosodic states for inter-F-segment interval. Two prosodic-state modeling schemes are proposed. One uses VQ to directly classify input features of two contiguous F-segments into 8 prosodic states. The other uses an RNN, trained with relevant linguistic features as output targets, to implicitly represent the prosodic status by the outputs of its hidden layer. Finite number of explicit prosodic states can then be obtained by vector-quantizing the outputs of the hidden layer of the RNN. In the following, these five parts are discussed in more detail.

2.1. Spectral feature extraction

A pre-processing was used to extract spectral features for RNN-based pre-classification. In the pre-processing, a short-time spectral analysis by 256-point FFT was performed for each of 200-sample frame padding with 56 zero-samples. The frame shift is 100 samples. The spectrum of each frame was compressed non-linearly into 20 frequency channels (in mel-scale) using a bank of 20 triangular windows. The energy spectrum was then log-compressed and cosine-transformed to calculate 12 mel cepstral coefficients. Besides, 12 delta mel cepstral coefficients and one delta log-energy were also calculated using a 7-frame window. So, there are in total 25 spectral features used in the pre-classification.

2.2. The RNN pre-classifier

The function of the RNN pre-classifier is to discriminate each input frame among three broad classes of syllable initial, syllable final, and silence. Fig. 2 shows the architecture of the RNN. It is a two-layer network with all outputs of the hidden layer being fed-back to itself as additional inputs [Elman 1990]. The RNN has a distinct property of using its hidden nodes to represent the contextual status of the previous inputs. It is therefore suitable for discriminating dynamic speech patterns [Elman 1991, Robinson 1994]. The RNN can be trained by the back-propagation (BP) algorithm with output targets being set according to the segmentation positions given by an HMM recognizer [Lee 1991].

2.3. FSM-based segmentation

The function of the FSM is to segment the input utterance into I-, F-, S-, and T-segments. The FSM is designed to conform to the phonetic structure of Mandarin base-syllables. Fig. 3 shows the state transition diagram of the FSM. To drive the FSM, all the three outputs of the RNN are compared with two threshold values, TH_L and TH_H. While one output is higher than TH_H and the other two are all lower than TH_L, the FSM moves into the corresponding stable state if it is a legal one. Otherwise, the FSM stays at T state. In this study, TH_L and TH_H were set to be 0.2 and 0.8, respectively. After obtaining the state sequence encoded by the FSM, we divide the input utterance into I-, F-, S- and T-segments.

2.4. Modeling feature extraction

We then extract 4 features relevant to prosody modeling for each F-type segment. They include three features representing the two ending points and the mean of the pitch frequency contour overlapping with the current F-segment and one feature representing log-energy mean of the F-segment. There are in total 9 modeling features used for detecting the prosodic state of an inter-F-segment interval. They include the 8 features of the two neighboring F-segments and one additional feature which represents the duration of the S-segment located between the

two F-segments.

2.5. Prosodic state classification

Two prosodic state classification schemes are proposed in this study. One uses VQ to encode the input modeling feature vector and directly classify it as the prosodic state associated with the encoded codeword. The other uses a two-layer RNN, trained with appropriate linguistic features extracted from the associated text as output targets, to implicitly represent the prosodic status of the current inter-F-segment interval by using the outputs of its hidden layer. The RNN has the same structure shown in Fig. 2. The inputs of the RNN consist of the same 9 modeling features used in the first scheme. And 6 output linguistic features are used in this study. Two indicators showing, respectively, whether the current inter-F-segment interval is an inter-word boundary and an intra-word boundary; Two indicators showing whether the current inter-F-segment interval is a left boundary and a right boundary of a long word with length greater than or equal to 3 syllables; One indicator showing whether there exists a punctuation mark (PM) in the current inter-F-segment interval; One indicator showing whether the two neighboring F-segments belong to the same syllable. The prosodic state is finally obtained by vector quantizing the outputs of its hidden layer. Here, the codebook size is also set to 8.

3. Simulations

Effectiveness of the proposed method was examined by simulations. A continuous-speech Mandarin database provided by the Telecommunication Laboratories was used. The database contains 452 sentential utterances and 200 paragraphic utterances. Texts of these 452 sentential utterances are well-designed, phonetically-balanced short sentences with lengths less than 18 characters. Texts of these 200 paragraphic utterances are news selected from a large news corpus to cover a variety of subjects including business, medicine, social event, sport, literature, etc. All utterances were generated by a single male speaker. They were all spoken naturally at a speed of 3.5-4.5 syllables per second. The database was divided into two parts: a training set and an open test set. These two sets consist of 28060 and 7034 syllables,

respectively.

We first examine the performance of the RNN pre-classifier and the FSM-based segmentation. Figs. 4(a-e) shows a typical example. It can be seen from these figures that all syllable finals have been properly detected. Each syllable final has an F-segment associating with it. By carefully examining all segmentation results, we find that the error rate for syllable-final segmentation is about 7%. Two major types of error had been found. One is the missing of an F-segment, i.e., a syllable final has no F-segments assigned to it. This usually occurred for syllables with Tone 5. Another type of error is caused by the missing of an I-segment to make two neighboring finals being connected together to form a long aggregate F-segment. This usually occurred at short voiced initials including nasal, liquid, and null initials. It is noted that these errors are relatively unimportant to our prosodic-state detection because our purpose is to model the global characteristics of the prosody-phrase structure. No significant cues of the prosodic phrase structure are lost due to these errors. So both the RNN and the FSM functioned quite well to meet our requirement [Chen 1996b].

We then examine the performance of the first prosodic state classification scheme using VQ to encode the modeling features. Fig. 5 shows the codewords associated with the 8 prosodic states. By examining the pitch parameters of these codewords, we find that a meaningful mapping between these prosodic states and the prosodic phrases structure does exist. Specifically, the first two states correspond to the beginning part of a prosodic phrase. States 6 and 8 correspond to the ending part. States 4, 5 and 7 correspond to the intermediate part. States 3 and 6 correspond, respectively, to minor and major breaks. Table 1 lists some statistics, including the distributions of beginning and ending of sentential and paragraphic utterances, and the distribution of the existence of a PM in the current inter-F-segment interval. It can be seen from this table that most utterances begin with State 1 and end with State 6. And most PMs are associated with State 6. Table 2 lists the state transition probabilities. It can be found from the table that States 2 and 4 follow State 1 to locate in the beginning part of a prosodic phrase. State 5 follows States 2 and 4. State 7 follows State 4 and 5. States 6 and 8 follow State 7. Fig. 4(h) shows a typical example of the encoded state sequence of an input utterance. And Fig. 6 shows a finite state automata (FSA) obtained by drawing only significant transition probabilities. Based on above discussions, the FSA is a meaningful

model to describe the prosodic phrases structure of Mandarin speech [Wang 1994, Chen 1996a, Elman 1990, Elman 1991].

Lastly, we examine the performance of the second RNN-based prosodic-state-detection scheme. The same statistics and state transition probabilities were calculated and listed in Tables 3 and 4. It can be seen from Table 3 that almost all utterances start with State 7 and end with State 2. And most PMs are associated with State 2. As compared with the results shown in Tables 1 and 2, the RNN-based method performed better on prosodic-state detection than the VQ-based method. Similar FSA can also be obtained by counting only significant state transition probabilities (see Fig. 7). Fig. 4(h) shows a typical example of the encoded state sequence of an input utterance. Finally, we examine the outputs of the RNN. Fig. 4(g) shows the outputs of inter-word and intra-word indicators. It can be seen from the figure that significant inter-word response always occurs at a word boundary. This property might be useful to assist in speech recognition.

4. Conclusions

A new prosody modeling method has been discussed in this paper. Experimental results have shown that a meaningful mapping between the resulting detected prosodic states and the prosodic phrase structure can be found. So its performance is quite well. Due to its effectiveness, further studies to incorporate it into a conventional speech recognizer is worthy doing in the future.

Acknowledgment

This work was supported by the National Science Council, Taiwan, ROC under the contract NSC-86-2213-E009-097.

Reference

Chen S. H., Hwang S. H., and Wang Y. R. (**1996a**), ''An RNN-based Prosodic Information Synthesizer for Mandarin Text-to-Speech'', Accepted by *IEEE Trans. on Speech and Audio Processing.*

Chen S. H., Liao Y. F., Chiang S. M., and Chang S. (**1996b**), ``An RNN-Based Pre-classification Method for Fast Continuous Mandarin Speech Recognition'', Accepted by *IEEE Trans. on Speech and Audio Processing.*

Compbell N. (**1993**), ``Automatic Detection of Prosodic Boundaries in Speech,'', *Speech Communication*, Vol.13, pp.343-354.

Elman J. L. (**1990**), ``Finding Structure in Time'', *Cognitive Science*, Vol.14, pp.179-211.

Elman J. L. (**1991**), ``Distributed Representations, Simple Recurrent Networks, and Grammatical Structure'', *Machine Learning*, Vol.7, pp.195-224.

Kompe R., Kiebling A., Niemann H., Noth E., Schukat-Talamazzini E.G., Zottmann, A. and Batliner A. (**1995**), ``Prosodic Scoring of Word Hypotheses Graphs,'' in *Proc. EUROSPEECH,* pp.1333-1336.

Lee S. J., Kim K. C., Yoon H. and Cho J. W. (**1991**), ``Application of fully recurrent neural networks for speech recognition", *Proc. IEEE Intern. Conf. Acoust., Speech, Signal Process. (ICASSP)*, pp. 77-80.

Robinson A. J. (**1994**), ``An application of recurrent nets to phone probability estimation'', *IEEE Trans. on Neural Networks*, Vol. 5, No. 2, pp. 298-305, March.

Sagisaka Y., Campbell N. and Higuchi N. edited (**1996**), ``Computing Prosody - Computational Models for Processing Spontaneous Speech'', *Springer-Verlag*, New York, Inc..

Wang Y. R. and Chen S. H. (**1994**), ``Tone recognition of continuous Mandarin speech assisted with prosodic information", *J. Accoust. Soc. Am.*, **96** (5), Pt. 1, pp. 2637-2645, Nov..

Wightman C. W. and Ostendorf M. (**1994**), ``Automatic Labeling of Prosodic Patterns,'', *IEEE Trans. Speech and Audio Proc.*, Vol.2, No.4, pp.469-480, Oct..

State / Prob. / Prosodic cue	1	2	3	4	5	6	7	8
Beginning of utterance	0.64	0.17	0.13	0.03	0.03	0.00	0.00	0.00
Ending of utterance	0.03	0.01	0.06	0.01	0.01	0.85	0.00	0.03
PM	0.04	0.02	0.12	0.02	0.08	0.63	0.01	0.08

Table 1: The statistics of the prosodic states detected by the VQ-based scheme.

State / Prob. / Prosodic cue	1	2	3	4	5	6	7	8
Beginning of utterance	0.00	0.00	0.03	0.00	0.01	0.00	0.96	0.00
Ending of utterance	0.08	0.92	0.00	0.00	0.00	0.00	0.00	0.00
PM	0.17	0.73	0.04	0.02	0.01	0.01	0.00	0.01

Table 3: The statistics of the prosodic states detected by the RNN-based scheme.

Next state / Prob. / Previous state	1	2	3	4	5	6	7	8
1	0.38	0.37	0.02	0.21	0.01	0.00	0.01	0.00
2	0.02	0.31	0.13	0.24	0.22	0.00	0.08	0.00
3	0.31	0.37	0.01	0.30	0.00	0.00	0.01	0.00
4	0.00	0.00	0.14	0.00	0.40	0.10	0.21	0.16
5	0.00	0.08	0.17	0.10	0.38	0.06	0.20	0.01
6	0.47	0.27	0.11	0.13	0.02	0.00	0.00	0.00
7	0.00	0.00	0.05	0.00	0.04	0.35	0.08	0.48
8	0.00	0.03	0.13	0.05	0.27	0.18	0.19	0.16

Table 2: Prosodic state transition probabilities of the VQ-based scheme.

Next state / Prob. / Previous state	1	2	3	4	5	6	7	8
1	0.00	0.01	0.22	0.18	0.08	0.04	0.32	0.14
2	0.00	0.00	0.06	0.02	0.01	0.00	0.76	0.12
3	0.03	0.15	0.08	0.05	0.29	0.11	0.04	0.39
4	0.04	0.01	0.10	0.14	0.05	0.24	0.00	0.28
5	0.14	0.13	0.21	0.07	0.36	0.21	0.00	0.12
6	0.38	0.17	0.09	0.32	0.01	0.31	0.00	0.05
7	0.00	0.00	0.27	0.01	0.55	0.09	0.00	0.06
8	0.03	0.20	0.19	0.29	0.00	0.27	0.00	0.02

Table 4: Prosodic state transition probabilities of the RNN-based scheme.

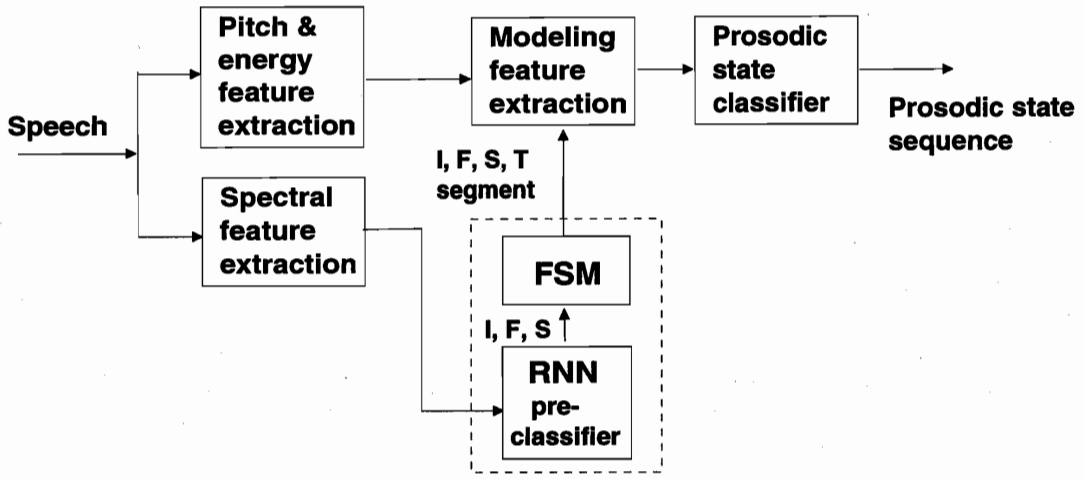


Figure 1: A block diagram of the proposed prosodic-state detection method.

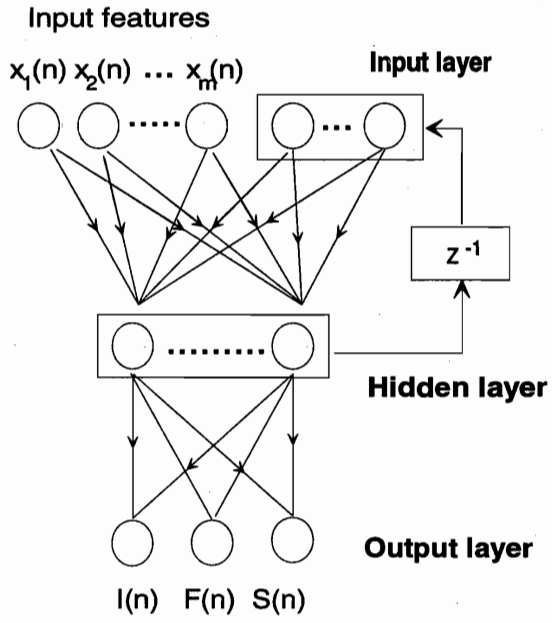


Figure 2: The architecture of the RNN.

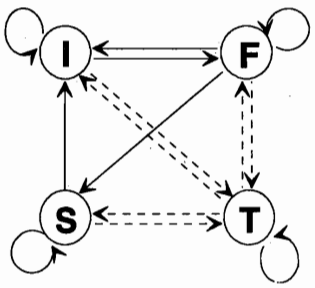
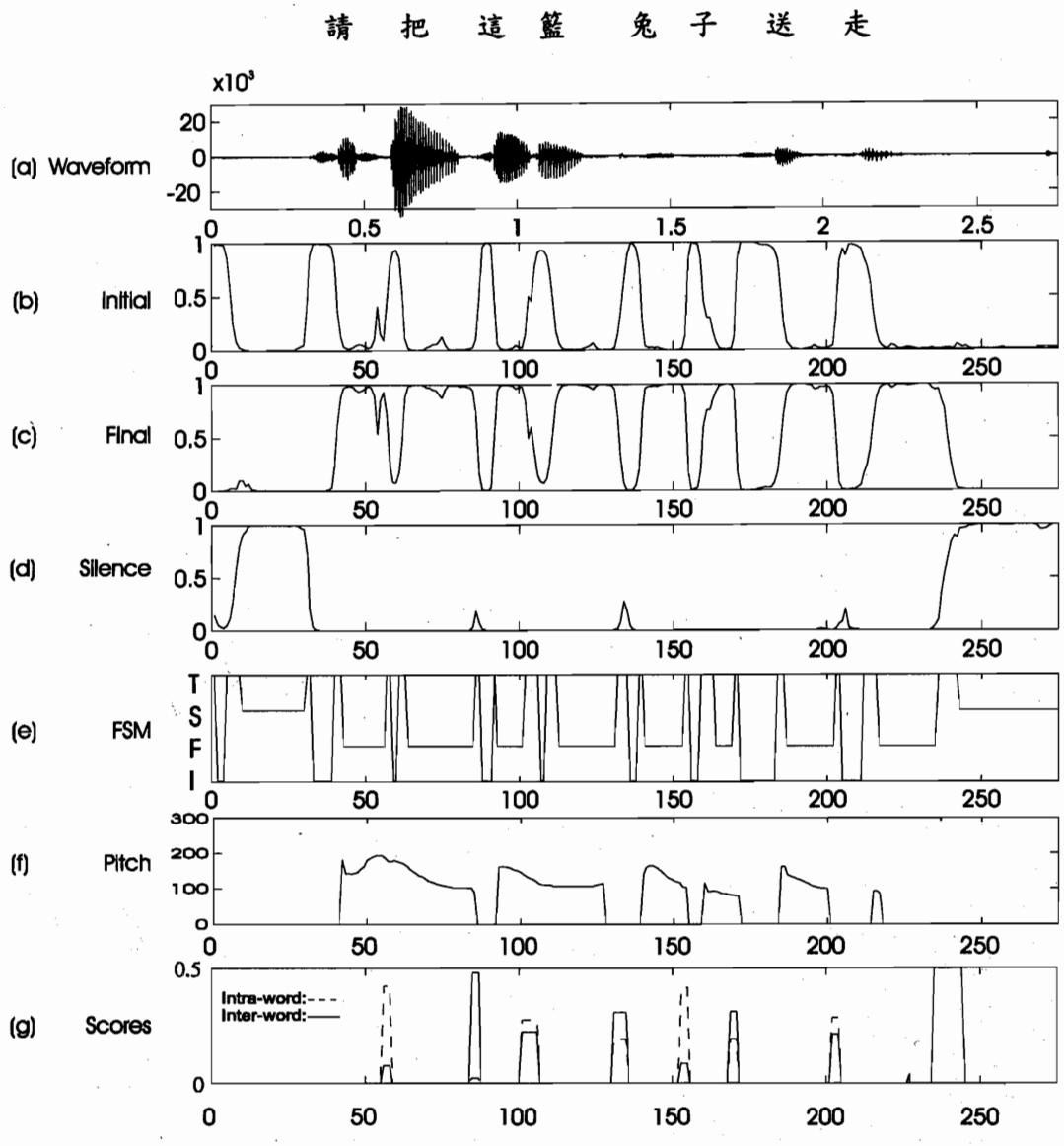


Figure 3: The state transition diagram of the FSM.



(h) Prosodic state sequence :

VQ :	4	5	4	5	7	8	7	6	
RNN :	7	3	8	4	8	4	8	2	

Figure 4: A typical example of the proposed mrthod : (a) Waveform of the input speech; (b) Initial-, (c) final- and (d) silence-outputs of the RNN pre-classifier; (e) Segmentation results of the FSM; (f) Pitch contour; (g) The inter- and intra-word scores generated by the RNN-based prosodic-state classifier; (h) Prosodic state sequences generated by the VQ-based and RNN-based schemes.

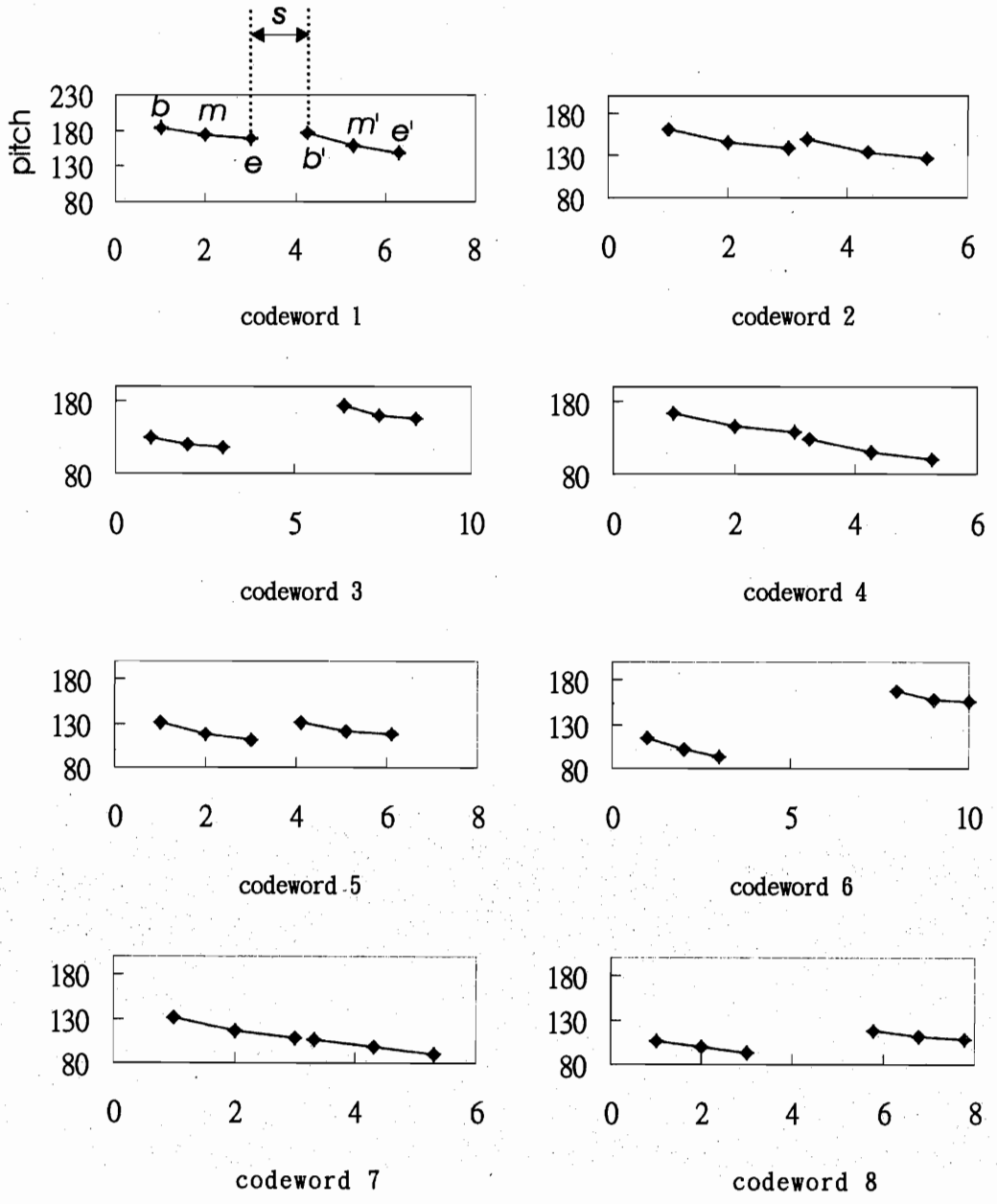


Figure 5: VQ codewords associated with the 8 prosodic states, where *b,b'* are the beginning points of F0 contours, *e, e'* are the ending points of F0 contours, *m, m'* are means of F0 contours, and *s* is the average duration of S-segment.

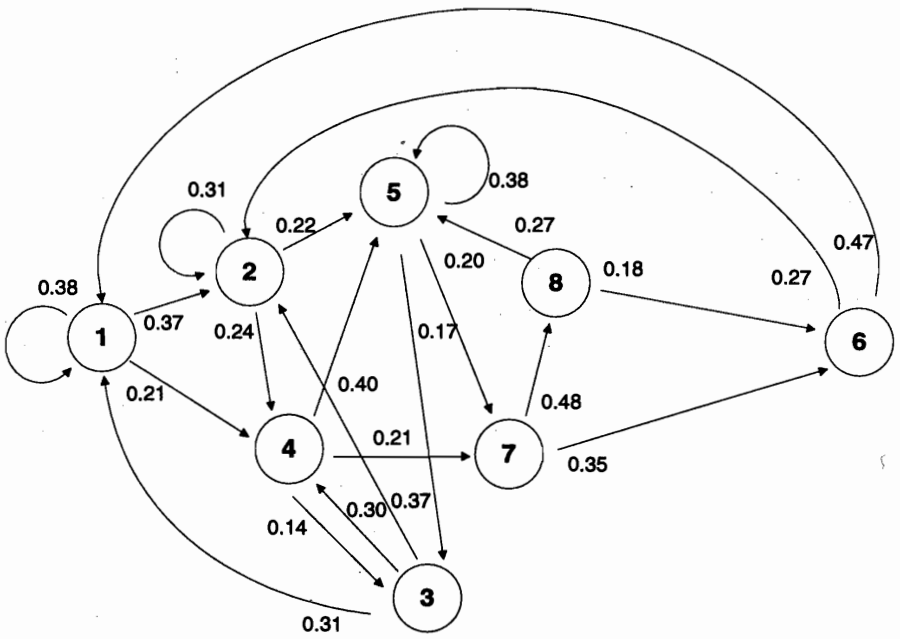


Figure 6: A FSA obtained by the VQ-based prosodic modeling scheme.

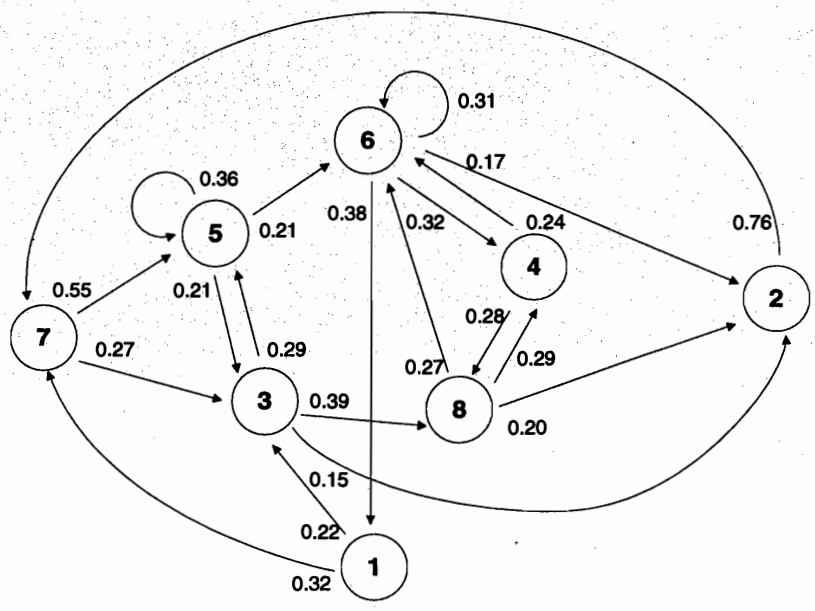


Figure 7: A FSA obtained by RNN-based prosody modeling scheme.

Rejection in Speech Recognition Based on CDCPMs

Mingxing Xu, Fang Zheng, Wenhu Wu
Speech Lab, Dept. of Comp. Sci. & Tech, Tsinghua Univ., Beijing 100084, China
[xumx, fzheng]@sp.cs.tsinghua.edu.cn, fzheng@cenpok.net

Abstract

Rejection is important for two-stage speech recognition. In this paper a new rejection method based on Center-Distance Continues Probability Model (CDCPM) is proposed, named CAP, which is the feature percentage in critical area (CAP) according to the probability theories. Also another rejection method named recognition score gap (RSG) is proposed to cooperate with CAP. Experiments are done across a large real-world database with 20,000 test samples. The average recognition accuracy is 86.33% with 3.46 candidates number on an average.

1. Introduction

In two-stage speech recognition or keyword spotting systems, rejection is a very important stage. In the first-stage, often as many candidates as possible are selected so that the correct candidate is contained and the accuracy is guaranteed. In the acceptation/rejection stage, efficient methods should be adopted to reject false hits in order to lower down the false alarm rate.

In many speech recognition applications, such as keyword spotting, the acceptation/rejection is performed using statistical hypothesis testing [Rahim 1995, Rose 1995, Sukkar 1995]. Moreover, many of the proposed methods use the recognition models themselves in formulating the verification likelihood ratio. In such a case, the recognition models are used for both recognition and rejection, and recognition/rejection performance tradeoff have to be considered. And there are some applications formulate the rejection test by constructing and disciminatively training verification-specific models to estimate the distributions of the null and alternate hypotheses. All of these methods need extra modeling and training.

In this paper a new rejection method based on Center-Distance Continues Probability Model (CDCPM) [Zheng 1996] is proposed, named the feature percentage in the critical area (CAP). The parameters used in CAP are different from those in the first recognition stage. According to this method the acceptation/rejection stage evaluate each recognition candidate individually. In experiments, we find the correct candidate's position has relation with the distribution of the recognition scores of candidates, which introduces another rejection method named recognition score gap (RSG) to cooperate with CAP. To evaluate the rejection efficiency, we give four definitions, probability of correctness (PC), probability of occurrence (PO), average recognition accuracy (ARA) and average candidates number (ACN).

This paper is organized as follows. In section 2, the theories of CAPs are discussed. In section 3, the method RSG is discussed. In section 4, we discuss how to use CAP and RSG cooperatively. In section 5, the experimental results are analyzed . In section 6, the conclusion is presented.

2. The Feature Percentage in Critical Area (CAP)

The CDCPM is a modified version of HMM with left-to-right architecture [Yang 1995], which eliminates the initial probability distribution and the probability transition matrix. The feature space of each state is divided into several sub-spaces described by one Center-Distance Normal (CDN) distribution [Zheng 1997a]. These sub-spaces can be estimated by some clustering method according to some kind of criterion [Zheng 1997b].

For the Normal distribution

$$N(x;\mu,\sigma)=\frac{1}{\sqrt{2\pi}\sigma}e^{-(x-\mu)^2/2\sigma^2},\quad x\in(-\infty,\infty) \tag{1}$$

there are 95% samples fall into the critical area $[\mu-2\sigma,\mu+2\sigma]$. Similarly, for the normal-derived Center-Distance Normal (CDN) distribution

$$N_{CD}(x;\mu,\sigma)=\frac{2}{\pi D}e^{-(x-\mu)^2/\pi D^2},\quad x\in[0,\infty) \tag{2}$$

there will be about 95% samples' center-distances fall into the critical interval $[0, 2.5D]$ because $\sigma=\frac{\sqrt{2\pi}}{2}D\approx 1.25D$.

Hereafter, denote the model by $\Lambda=\{\mu_{nm},D_{nm}|1\le n\le N,1\le m\le M\}$ and the observation feature vector sequence by $O=\{o_1,o_2,\cdots,o_T\}$, where N is the number of states

in the model, M is the number of densities in a model state and T is the utterance length in frame, three CAP rejection methods are discussed in details [Zheng 1997c].

2.1 CAP1

Define the acceptation/rejection score (ARS) for any feature vector o_t as

$$Score(o_t|\Lambda) = \begin{cases} 1, & \max\limits_{1\le m\le M}\left\{d(o_t,\mu_{nm})\middle| o_t \in Density(n,m)\right\} \in [0, kD_{nm}] \\ 0, & otherwise \end{cases} \tag{3}$$

In Equ. (3), $Density(n,m)$ denotes the m-th density in n-th state for the model Λ and $d(\cdot,\cdot)$ is the distance measure for feature vectors. The critical area controlling parameter k=2.5 (or other value). Based on Equ. (3), the ARS of feature sequence O with model Λ is defined as

$$Score(O|\Lambda) = \frac{1}{T}\sum_{t=1}^{T} Score(o_t|\Lambda) \tag{4}$$

Obviously, this definition satisfies the limitation $Score(O|\Lambda) \in [0,1]$. And if we set the acceptation/rejection thresholds such as $TSH_h \ge 0.5$ and $TSH_l \le 0.5$, the category of O will be determined by

$$O\begin{cases} \in \Lambda, & \text{if } Score(O|\Lambda) > TSH_h \\ \notin \Lambda, & \text{if } Score(O|\Lambda) < TSH_l \end{cases} \tag{5}$$

The thresholds can be chosen empirically or by the analysis of the training data.

2.2 CAP2

In Equ. (2), the ARS is related only to the maximally matched density, if all densities are considered, the vector ARS can be defined as

$$Score(o_t|\Lambda) = \begin{cases} 1, & \sum\limits_{m=1}^{M} Score\left(o_t\middle|Density(n,m)\right) > TSH_{NUM} \\ 0, & otherwise \end{cases} \tag{6}$$

where

$$Score(o_t|Density(n,m)) = \begin{cases} 1, & d(o_t,\mu_{nm})\middle| o_t \in Density(n,m)) \in [0, kD_{nm}] \\ 0, & otherwise \end{cases} \tag{7}$$

and TSH_{NUM} is a number ranging from 1 to M-1.

Equ.s (6), (7) with (4), (5) gives another CAP acceptation/rejection quantity which is different from CAP1 only in that CAP1 considers the nearest density while CAP2 considers all densities inside one state.

2.3 CAP3

In CAP1 and CAP2, the ARS of each feature vector is a two-value function. Our experiments show that different number of densities for different states performs better [Zheng 1997b]. In that situation, CAP1 and CAP2 can not reflect the differences. We think it better to consider all the 2-value scores for the feature vector with every density. Thus the ARS for the sequence O with the given model is defined as

$$Score(O|\Lambda) = \frac{\sum_{t=1}^{T} \sum_{m=1}^{M(n(t))} Score(o_t | Density(n(t), m))}{\sum_{n=1}^{N} M(n)} \tag{8}$$

where $n(t)$ denotes the state that the t-th feature vector belongs to and $M(n)$ is the density number in state n.

Equ. (8) leads to CAP3, where there is no need to set up TSH_{NUM} thresholds for all states as in CAP2.

3. The Recognition Score Gap (RSG)

The matching score provided by recognition module indicates how the utterance matches the model. In order to include the correct result, the first-stage recognition module often outputs the K best candidates. The scores of Top K candidates contain the information of the position of the correct answer. In our experiments, we find the score differences between adjacent candidates are useful for the acceptation/rejection stage.

At first we calculate the mean and variance of the candidates' scores, then we look for the relation between these values and the position of correct candidate. The experimental results show that a large score gap appears between the right candidates and wrong ones, as shown in figure 1.

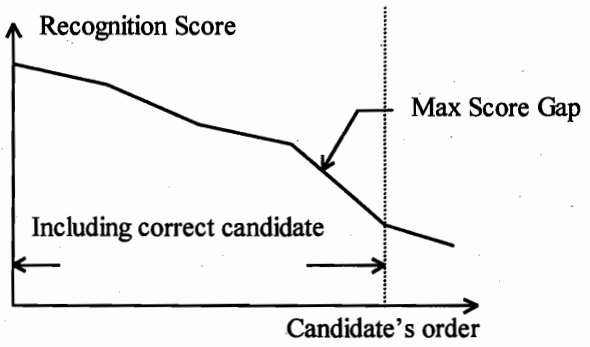


Figure 1 The Curve of Recognition Scores

A specified threshold is used to determine the number of reserved candidates, only those with score higher than the threshold are reserved.

Let $RS(k)$ denotes the recognition score of the k-th candidate, $k = 0,1,\cdots,K-1$, where K is the number of candidates that the first-stage module outputs. Define the Recognition Score Gap (RSG) for the k-th candidate as

$$RSG(k) = RS(k) / RS(0) \tag{9}$$

Setting up a threshold TSH for RSG, we can throw off those candidates whose RSG's are smaller than the given threshold as

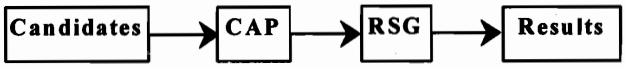
$$k\begin{cases} \in \Lambda, & \text{if } RSG(k) \geq TSH \\ \notin \Lambda, & \text{if } RSG(k) < TSH \end{cases} \tag{10}$$

4. Using Several Rejection Methods Jointly

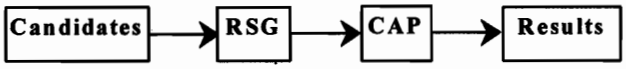
In above rejection methods, CAPs calculate rejection score for each candidate independently. In other word the result score is only dependent on the utterance's feature vector sequence. Clearly this kind of rejection operation is a filtration procedure that will perhaps output no candidates. Whereas the RSG method computes the rejection score according to the relation among candidates provided by recognition module at the first stage. Usually these candidates are sorted by recognition scores. We use RSG method to delete candidates from the rejection point in the candidates list. So the result of this kind of rejection operation is just like a bobtail with the first candidate at least.

Because CAP and RSG are based on different theories, there is no correlation between these two rejection methods. We can use them jointly. For example, we can:

(1) use CAP at first, then RSG, as



(2) or use RSG at first, then CAP as



5. Experimental Results

Some experiments have been done across a real-world spontaneous database. The speech data are taken from telephone network and sampled at 8KHz. The samples are 13-bit linear PCMs expanded from A-law codes. The database consists of speech data uttered by 200 people, and the amount is about 4GB. 10^{th} order mel-frequency cepstral (MFCC) analysis is performed on 32 ms speech window every 16 ms. Auto-regressive analysis is also performed on 5 adjacent frames of MFCC vectors. The MFCCs and their corresponding auto-regressive coefficients are the features used for the CDCPM [Zheng 1996, 1997a] in this paper. The SRUs are 419 Chinese syllables. The first stage outputs 10 candidates sorted according to recognition scores to the next stage. We test 20,000 utterances. The experimental result is shown in figure 2.

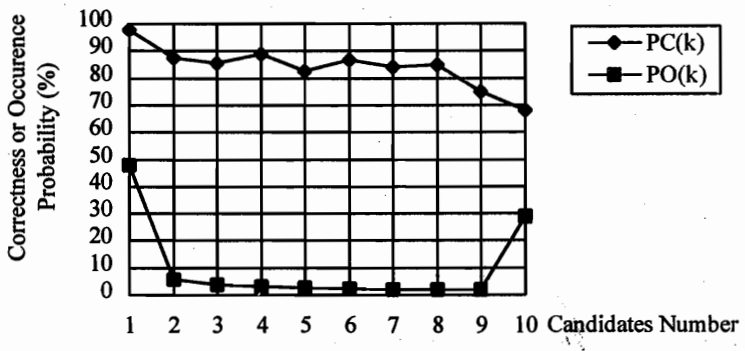


Figure 2 Rejection Results

The number of rejected candidates is 110,837. There are 302 correct candidates in them. The error rate is 0.27%.

In order to evaluate the performance of the rejection methods, we will give some definitions. Denote the total number of testing samples (e.g. 20,000) by TN, the total number of samples where k candidates are outputted in the acceptation/rejection stage by $C(k)$, and the total number of samples where k candidates including the correct one are outputted in the acceptation/rejection stage $R(k)$. Obviously $R(k)$ samples must be included in those $C(k)$ samples. Definitions are given as follows.

(1) Probability of Correctness is defined as

$$PC(k) = \frac{R(k)}{C(k)} \tag{11}$$

which specifies the correctly detection probability conditioned on k candidates are outputted.

(2) Probability of Occurrence is defined as

$$PO(k) = \frac{C(k)}{TN} \tag{12}$$

which indicates the probability of outputting k candidates, where

$$\sum_{k} PO(k) = 1 \tag{13}$$

(3) Average Recognition Accuracy (ARA) can be calculated by

$$ARA = \sum_{k=1}^{10} PC(k) * PO(k) \tag{14}$$

which describes the total performance of rejection method.

(4) Average Candidates Number (ACN) can be calculated by

$$ACN = \sum_{k=1}^{10} k * PO(k) \tag{15}$$

which also describes the total performance of rejection method.

From the data in Figure 2, we can calculate ARA and ACN. The results are ARA = 86.33% and ACN = 3.46 < 4.

6. Conclusion

We introduce two new rejection methods, named CAP and RSG, to decrease the number of candidates given in the first recognition stage. We also define some parameters to evaluate the total performance of rejection method. The experimental results show that CAP and RSG can provide significantly better performance. They have following four features:

(1) CAP and RSG are easy to calculate without extra training and modeling as in some other rejection methods.

(2) The *ACN* has been decreased to 4 after rejection procedure, which indicates the rejection method proposed is an efficient method.

(3) The *PO* is a U-type curve. This typical polarization feature fits the distribution of correct candidates in recognition results.

(4) The incorrectly rejection rate is as low as 0.27%, which shows the rejection methods we used can work well with CDCPM. This good performance also shows that the CDCPM can model Chinese speech well.

References

[1] **Rahim, M.G., Lee, C.-H., Juang, B.H.,** "Discriminative utterance verification for connected digits recognition," Proc. *Eurospeech'95*, pp. 529-532, Sept.1995

[2] **Rose, R.C., Juang, B.H., Lee, C.H.,** "A training procedure for verifying string hypotheses in continuous speech recognition", Proc. *ICASSP'95*, Vol. I, pp. 281-284, May 1995

[3] **Sukkar, R.A., Lee, C.H., Juang, B.H.,** "A vocabulary independent discriminatively trained method for rejection of non-keywords in subword-based speech recognition", Proc. *Eurospeech'95*, pp. 1629-1632, Sept.1995

[4] **Zheng, F., Wu, W.-H., Fang, D.-T.,** "CDCPM with its application in speech recognition," *(Chinese) J. of Software*, 7: 69-75, Oct. 1996

[5] **Yang, X.-J., et al** Speech Digital Signal Processing. Beijing: Electronic Industry Publishing House, 1995 (in Chinese)

[6] **Zheng, F., Chai, H.-X., Shi, Z.-J., Wu, W.-H., Fang, D.-T., (1997a)** "A real-world speech recognition system based on CDCPMs," *Int'l Conf. on Computer Processing of Oriental Languages (ICCPOL'97)*, 1: 204-207, Apr. 2-4, 1997, Hong Kong

[7] **Zheng, F., Xu, M.-X., Wu, W.-H., (1997b)** "The description of the intra-state feature space". Somewhere in this ROCLing X proceeding.

[8] **Zheng, F., (1997c)** "Studies on approaches of keyword spotting in unconstrained continuous speech," Ph.D. Dissertation. Beijing: Dept. of Comp. Sci. & Tech., Tsinghua Univ., 1997