RC22635 (W0204-131) April 29, 2002
Computer Science

IBM Research Report

Identification of Probable Real Words: An Entropy-Based Approach

Youngja Park
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Identification of Probable Real Words: An Entropy-based Approach

Youngja Park
pyoungja@us.ibm.com
IBM T.J. Watson Research Center
P.O. Box 704, Yorktown Heights
New York 10598, USA

Abstract

This paper proposes a method for identifying probable real words among out-of-vocabulary (OOV) words in text. The identification of real words is done based on entropy of probability of character trigrams as well as the morphological rules of English. It also generates possible parts-of-speech (POS) of the identified real words on the basis of lexical formation rules and word endings. The method shows high performance both in precision and in recall. This method is very useful in recognizing domain-specific technical terms, and has successfully been embedded in a glossary extraction system, which identifies single or multi word glossary items and builds a domain-specific dictionary.

1 Introduction

No lexicon could be expected to contain entries for every possible word of a language, given the dynamic nature of language and the creativity of human beings. Nowadays, this phenomenon has become even more challenging as new technologies develop faster than before. Thus, inevitably, there always exist out-of-vocabulary words[1] in documents. Especially, new derived words, such as new compound words and morphological variations of existing words (mostly by means of affixation), and technical words can be missing from a given lexicon. Words unknown to the lexicon cause a lot of problems to NLP systems which depend heavily on lexical information such as POS taggers and parsers.

There has been a great effort to address this problem, especially in the areas of POS taggers (Brill, 1995; Dermatas and Kokkinakis, 1995; Weischedel et al., 1993) and speech recognition (Gallwitz et al., 1996; Hazen and Issam, 2001). However, previous approaches begin the process based on the assumption that the out-of-vocabulary words are just unknown to the systems' lexicons but they are possible real words of the language. Then, they guess the most probable POS or the closest substitute of an unknown word. However, not every out-of-vocabulary word is a possible real word of the language. We have analyzed the out-of-vocabulary words in document collections from several domains, and found that only a small portion of the words are possible real words.

The goal of this work is recognizing real words among out-of-vocabulary words in text and finding lexical information of the words. This work has been motivated by our effort to build domain-specific glossaries (Park et al., 2002). While we were working on automatic glossary extraction, we noticed that technical documents contain a lot of words missing from a general-purpose dictionary, and many of them are actually important domain-specific words. However, only a few domains (for example, biomedical domain) have domain-specific

[1] words which are not found in a dictionary. Also called unknown words.

dictionaries available, and it is very difficult to obtain lexical resources for other domains. We concluded that the correct recognition of probable real words is very important not only to build a domain-specific dictionary but also to augment an existing general dictionary.

Based on the analyses of large collections of documents, we classify out-of-vocabulary words into the following categories.

- derived words
- new words
- proper nouns
- non-word strings

We address the problems of the recognition of real words and of guessing their POS on the basis of the types of out-of-vocabulary words.

Derived words are morphological variations of words already known to the lexicon, mostly by means of affixation, i.e., adding prefixes to the beginning of words or suffixes to the end, and by means of compounding, i.e., two or more words are written as one word (Pickett *et al.*, 1996). New words mean the words that can not be produced by the derivation (or word formation) rules from the existing words of the language. Many domain-specific technical terms belong to this category. Proper nouns are mostly person names and place nouns. We also consider upper case and non-initial mixed case words as proper nouns. Non-word strings mean alphabetic strings together with non-alphabetic characters such as numeric characters and other special characters. In this work, we don't take into account proper nouns and non-word strings because they are not valuable to be kept in dictionaries. Thus, in this work, out-of-vocabulary words are classified into two categories - derived words and new words.

The overall process for identifying real words and for producing lexical information is as follows. First, we remove all proper names and non-word strings from the document collection. It is easy to recognize non-word strings and upper case and non-initial mixed case words. In addition to the capitalization feature, we use a precompiled names database (Ravin et al., 1997) for recognizing person names and place names. If a word exists in the database, we consider it as a proper noun. Second, we look up all the remaining words in the document collection in a general purpose English dictionary built by IBM (IBM, 2001) and collect all the out-of-vocabulary words, i.e., words unknown to the dictionary, and their frequencies in the collection. Third, we discard words which appear only once in the collection. Fourth, we check if an out-of-vocabulary word is comprised of existing words in the dictionary and/or morphological units such as a prefix and a suffix. If this process succeeds, possible parts-of-speech of the word are generated based on the morphological rules applied to produce the word. Fifth, if this process fails, we judge if the word may be a new word on the basis of entropy of the probability of its character trigrams and guess its parts-of-speech from its ending characters.

The rest of this paper is structured as follows. We present morphological rule-based approach in section 2 and entropy-based approach in section 3. In section 4, we show experimental results and evaluate the performance of the proposed method. Previous related work is described in Section 5. Finally, we describe possible future improvements in section 6.

2 Morphological Rule Approach

This approach performs the recognition and POS guessing processes for out-of-vocabulary words given that the sub-components of the words are already known. There are three types of morphological variations - words with prefixes, words with suffixes, and compound words. For words with prefixes or suffixes, we use pre-collected sets of prefixes and suffixes for English - currently containing 75 prefixes and 76 suffixes. For compound words, we try to divide an out-of-vocabulary word into two possible existing words.

The process for prefixed words is as follows. First, the system checks if any of the prefixes in the prefix list appears at the beginning of the word. If a word contains a prefix, then the system chops the prefix off the word and looks up the remaining part (the root word) in the dictionary. We set the minimum length of a root word to two characters. If the dictionary contains the root word, the out-of-vocabulary word is regard as a real word, and the word inherits the lexical information of the root word. For example, *antiasthmatic*, *autoinjector*, *electrocardiography*, and *hypothyroidism* are discovered by the prefix process.

The processing for suffixes is more complicated. We have a rule set for suffixes, which describes the pre-conditional POS of a root word for having a specific suffix and the resulting POS condition. The suffix rule structure is as follows.

[*suffix*, { *precondition-POS* $\rightarrow$ *result-POS* }*]

For instance, the rule for suffix *able* is [*able*, {VB $\rightarrow$ JJ}, {NN $\rightarrow$ JJ}]. This means, a verb or a noun may have suffix *able* at the end of the word, and the resulting word's part-of-speech is an adjective. If a word contains a suffix, the system removes the suffix and recovers the root word. In English, when a suffix is added to a word, it may change spelling in the root word. For instance, words ending with a silent *e* usually drop the *e* before a suffix beginning with a vowel. An example of this case is *browsable*. The final *e* of *browse* was dropped as a result of adding *able*. Thus, after separating a suffix from the root word, we recover the original form of the root word by using linguistic information. If the recovered root word is found in the dictionary and it has one of the preconditioned POS, then the word is regarded as a real word and it has the result POS of the rule. Some examples of this case are *migranious*, *oxidizability* and *ventilatory*. Some words, for example, *remanufacturability*, may have a prefix and a suffix together. In this case, the word goes through both processes explained above.

If a word fails both the prefix processing and the suffix processing, it is considered for the compound processing. If a word consists of two content words which are known to the lexicon, and their parts-of-speech are one of the predetermined combinations, it is considered as a real word and has the second component's part-of-speech. The possible combinations of words are Noun+Noun, Noun+Participle form of verbs. If a word is composed of two words but does not belong to the possible combinations, the word is discarded. Some examples of the compound words are *eyedrops*, *photophobia*, *stereoselectivity*, *airbreathing*, and *doubleblinded*

3 Entropy Approach

3.1 Identification of new words

Human beings can very successfully guess whether a word never seen before is a possible real word or not, even though the word is not comprised of already known words. We assume that human beings may conclude that a word is a possible word of the language, if the character sequences in the word look probable, and it is natural to pronounce. We base the recognition of non-derivational new words on this assumption. That is, this method is based on the prediction of a language; how well can the next letter of a text be predicted when the preceding n letters are known (see, Shannon, 1951 for more extensive description of estimating the entropy and redundancy of a language).

A word is a cohesive group of letters with strong internal statistical influences (Shannon, 1951). We regard a string as a possible real word if every letter in the string is likely to co-occur with its neighbors. That is, if the letters in a word have high chances to occur in their position given the preceding characters have been seen (i.e., an n-gram model), the word is regarded as a real word. More formally, we compute entropy of the probabilities of n-gram sequences in an out-of-vocabulary word, and if the entropy value is high, we conclude the

word is a real word. In this work, the neighbors of a character are defined as the two preceding characters, i.e., a trigram model.

The probability of a character, c_3, given the two characters preceding it, c_1, c_2, is estimated as in Equation 1.

$$P(c_3|c_1, c_2) = \frac{f(c_1c_2c_3)}{f(c_1c_2)} \quad (1)$$

In this equation, $f(c_1c_2c_3)$ denotes the number of times the sequence of characters $\{c_1, c_2, c_3\}$ is observed and $f(c_1c_2)$ denotes the number of times the sequence $\{c_1, c_2\}$ is observed in a training corpus. In this work, we produce training data by generating all the possible forms (base forms and inflectional forms) of the words in our dictionary (IBM, 2001). The training data consist of 81,274 words.

To estimate the trigram probabilities, we add one leading space and one trailing space to every word, making a 28 letter alphabet. That is, for a word w with n characters, $c_1\, c_2 \cdots c_n$, we add a leading space (c_0), and an trailing space (c_{n+1}) and generate $w' = c_0\, c_1\, c_2 \cdots c_n\, c_{n+1}$. Then, we produce all the two character sequences, $c_0\, c_1, \cdots, c_n\, c_{n+1}$, and the three character sequences, $c_0\, c_1\, c_2, \cdots, c_{n-1}\, c_n\, c_{n+1}$, and count their frequencies in the training data. At last, we compute the probabilities of all the possible trigrams by using Equation 1 and store them in a look-up table.

To compute entropy of an out-of-vocabulary word, $w = c_1\, c_2 \cdots c_n$. we add a leading space (c_0) and a trailing space (c_{n+1}) to the word and divide it into trigrams and search each trigram in the look-up table. The entropy of a word w, $H(w)$, is defined as in Equation 2.

$$I(w) = -log_2\ P(c_i|c_{i-1},\ c_{i-2})$$

$$H(w) = \sum_{i=2}^{n+1} P(c_i \mid c_{i-1},\ c_{i-2}) I(w) \quad (2)$$

If a given word satisfies the following two conditions, it is regarded as a possible real word.

- the number of unknown trigrams[2] is less than a threshold value, θ_1
- the entropy of a word is greater than a threshold value, θ_2

Currently, θ_1 is set to 2 if the length of a word is less than or equal to 10 and set to 3 if the length is greater than 10. θ_2 is set to 2.3, which was determined from the average entropy minus the minimum entropy of the training data.

3.2 POS guessing for new words

In addition to identifying probable new words, this system produces possible parts-of-speech of the words. We adopt the ending guessing method described in Mikheev (1997) for this purpose. We collect the ending guessing rules from the training data described in section 3.1. For all the words in the training data, we generate all possible endings from length 1 up to length 5, together with the parts-of-speech of the words. We set the minimum length of the remaining part to 3. Table 1 shows how ending guessing rules are generated from our training data. Throughout this paper, POS tags are represented by Penn Treebank Tag code (Marcus and Santorini, 1993).

word	ailments	mounting	
Ending Rules	ments NNS	nting NN	nting VBG
	ents NNS	ting NN	ting VBG
	nts NNS	ing NN	ing VBG
	ts NNS	ng NN	ng VBG
	s NNS	g NN	g VBG
word	abandons	primary	
Ending Rules	ndons VBZ	mary NN	mary JJ
	dons VBZ	ary NN	ary JJ
	ons VBZ	ry NN	ry JJ
	ns VBZ	y NN	y JJ
	s VBZ		

Table 1: Examples of Ending Guessing Rules

All the ending rules and their frequencies are collected from the training data, and infrequent rules (frequency = 1) are discarded from the rule set. The rule set contains 12,387 rules, and the most frequent 50 rules are as shown in Table 2. The numbers in parentheses denote the frequencies of the rules.

[2]trigrams which are not shown in the training data

s NNS (19301)	g VBG (7116)	ng VBG (7111)
ing VBG (7075)	s VBZ (7006)	d VBD (6879)
ed VBD (6768)	es NNS (5119)	y RB (4701)
ly RB (4668)	rs NNS (4336)	r NN (4315)
e NN (3562)	ers NNS (3462)	es VBZ (3436)
er NN (3423)	s NN (3423)	ss NN (3179)
ess NN (3136)	ness NN (3068)	n NN (2843)
e JJ (2773)	e VB (2763)	t NN (2498)
y NN (2490)	ts NNS (2422)	on NN (2026)
ns NNS (1988)	t JJ (1858)	ion NN (1803)
d JJ (1725)	tion NN (1569)	ting VBG (154)
ted VBD (1482)	l JJ (1451)	ed JJ (1408)
ons NNS (1391)	ies NNS (1330)	le JJ (1323)
y JJ (1297)	ble JJ (1238)	lly RB (1237)
al JJ (1212)	ation NN (1187)	ions NNS (1185)
ty NN (1184)	r JJ (1152)	ity NN (1084)
ally RB (1075)	able JJ (1056)	

Table 2: Frequent Ending Guessing Rules

By using the rule set, the system produces all possible parts-of-speech of a word on the basis of the longest matching pattern. We look up the ending letters of the word in the rule set from the longest ending (5 letters if the word's length is larger than 7, otherwise the word's length minus 3) to the ending of length 1 (the final letter). If an ending exists in the rule set, the matching process stops, and the system produces all the parts-of-speech of the ending in the order of the rule frequencies. For instance, *cortical* is guessed as an adjective and a noun, but adjective reading is preferred because *tical* appears 105 times as an adjective and 4 times as a noun in the training data.

4 Experiments and Results

In order to evaluate the performance of the proposed method, we ran the algorithm on a collection of 4,000 MEDLINE abstracts. MEDLINE is an on-line computer database of abstracts and references from biomedical journals[3]. The collection consists of about 900,000 words, and the file size is about 6.9 megabytes. The main reason we selected medical documents for the experiment is many of the medical terms are not included in general-purpose language dictionaries, but we can relatively easily verify these words because many medical dictionaries have been built, and some of them are available on the web.

[3]http://www3.ncbi.nlm.nih.gov/

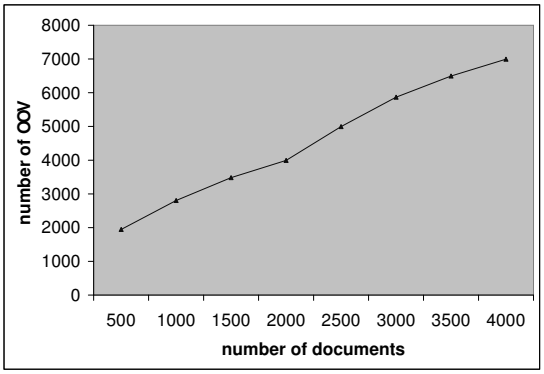

Figure 1: The number of out-of-vocabulary words in the MEDLINE collection

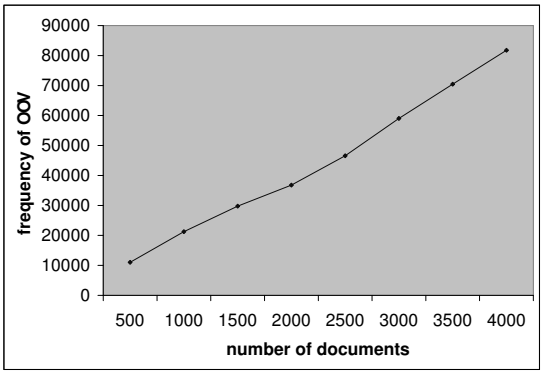

Figure 2: The frequencies of out-of-vocabulary words in the MEDLINE collection

The system found a total of 6,997 out-of-vocabulary words from the MEDLINE collection. Figure 1 shows the number of out-of-vocabulary words, and Figure 2 shows the frequencies of these out-of-vocabulary words in the MEDLINE collection. As we can see from the pictures, the number of out-of-vocabulary words and their uses monotonically increase in proportion to the increase of the document size. We excluded words with frequency 1 in the collection from consideration, resulting in a total of 4,187 words. The detailed types and the frequencies of the extracted out-of-vocabulary words are shown in Table 3.

The system decided 2,815 words among 4,187 out-of-vocabulary words are probably

OOV Type	*Count*
Affixation	443
compound	85
new word	2287
Misc.	1372

Table 3: Types of out-of-vocabulary words in the MEDLINE collection

real words, and 1,372 words are not real words (see, Table 4). In order to verify the system's judgment, we have conducted two verification processes. At first, we looked up all out-of-vocabulary words in a medical dictionary. We used the on-line version of the Merriam-Webster medical dictionary[4] for this purpose. Then, for words which do not exist in the medical dictionary (we assume they are mostly non-medical words), the human judges (non-domain experts) decided whether they are probable English words or not by referencing an on-line English dictionary[5]. If an out-of-vocabulary word appears in one of the two dictionaries, we regard it as a real word.

We developed a Perl (Wall and Schwartz, 1992) script program to automate the dictionary look-up processes. This program performs dictionary look-up with a URL and a word without any human intervention. It accesses a webpage of the given URL, and performs search with the given word, and returns the webpage of the search result. Then, it parses the returned webpage and decides if a word was found or not.

Table 4 shows the result of this experiment. The first column (*Dictionary Lookup-Yes*) denotes the number of the words found in one of the dictionaries, and the second column (*Dictionary Lookup-No*) denotes the number of the words which don't exist in any of the dictionaries. The first row (*System Guess-Yes*) denotes the number of the words which the system considered as real words, and the second row (*System Guess-No*) is the number of the words which the system regarded as invalid words.

[4] http://www.intelihealth.com/IH/ihtIH/WSIHW000/9276/9276.html

[5] http://www.dictionary.com

		Dictionary LookUp		
		Yes	No	sum
System Guess	Yes	2341	474	2815
	No	579	793	1372
	sum	2920	1267	4187

Table 4: Performance of Experiments

The performance of this system on the medline collection is as follows.

$precision = 83.16\%$

$recall = 80.17\%$

$F - measure = 81.64\%$

However, many of the samples that the system decided real words but were not found in the dictionary (*System Guess-Yes* and *Dictionary Lookup-No*) are actually real words. This is because the dictionary used for this experiment is also limited. Some examples of the words – mostly biology terminology and drug/treatment names – are *aggregometry*, *cardiomyocyte*, *colforsin*, *nondihydropyridine*, *nylestriol*.

5 Related Work

There has been a great effort to address this problem, especially in the areas of POS taggers and speech recognition. However, different applications recognize the problem of out-of-vocabulary words in different perspectives and have different goals. For POS taggers and parsers, which rely on lexical (syntactic) information about words, the goal is to guess the most plausible part-of-speech and other lexical information of an out-of-vocabulary word in a context. Dermatas and Kokkinakis (1995) estimated the probability that an unknown word has a particular POS tag from the probability distribution of words which occur only once in the previously seen texts. More advanced POS guessing methods use leading and trailing word segments to determine possible tags for unknown words. Weischedel et al. (1993) proposed a POS guessing method for unknown words by using the probability for an unknown word to be of a particular POS tag, given its capitalization feature and its ending.

Brill (1995) describes a system of rules which uses both end-guessing and more morphologically motivated rules. Mikheev (1997) presents a technique for fully automatic acquisition of rules which guesses possible POS tags for unknown words using their starting and ending segments. For speech recognition systems, an out-of-vocabulary(OOV) word is either a word unknown to the system vocabulary or a word that the recognizer fails to recognize. The goal is to find the closest word (in terms of sound and meaning) to the OOV word from the system's vocabulary.

Character ngram-based statistical approaches have been used in word-level language processing such as spelling correction (Angell et al., 1983), word segmentation (Juola et al., 1994), and language identification (Dunning, 1994). Angell, Freund and Willett (1983) describe a method of comparing misspellings with dictionary terms based on the number of trigrams that the two strings have in common, using Dice's similarity coefficient as the measure of similarity. The misspelled word is replaced by the word in the dictionary which best matches the misspelling. Juola, Hall and Boggs (1994) describes a system which segments full words into their constituent morphemes based on entropy of the probabilities of trigram sequences. Dunning (1994) implements a high accuracy language identification using character n-gram models and a Bayesian classifier. The performance of the n-gram language classifier is evaluated using different size of n-grams.

6 Conclusions

We have developed an approach to augmenting a morphological lexicon with new words such as newly derived words and domain-specific technical words through text analysis of document collections. For morphologically derived words, we have employed morphological rule-base methods such as affixations and compound words. We have also proposed a new technique to identify non-derivational new words based on entropy of the probabilities of trigram sequences. The probabilities of trigram sequences are trained on an existing English dictionary.

Some possible improvements and future plans are as follows.

1. The rules used for compound word process is over-generalized. For example, all the combinations of two nouns may not be compound nouns. We anticipate the performance will be improved if we incorporate a corpus statistics-based compound word processing scheme into the existing method.

2. The performance will be improved if we include domain-specific prefixes and suffixes such as *amino*, *brancho*, and *cardio*.

3. We trained our entropy model on a general-purpose dictionary. However, many medical and biotechnological terms have their origin in Latin. We expect the performance would be better if we train our system by using a domain-specific lexicon or a tagged (specified if a word is correct or not) domain-specific corpus.

4. We expect it is not difficult to apply this approach to other languages because this system only uses basic morphological rules of a language and language-independent statistical information. In addition, it does not require a large amount of annotated training data.

Acknowledgement

I am grateful to Marco Gruteser for the development of the Perl script for the evaluation of this work. I also thank anonymous reviewers for their helpful comments.

References

R. Angell, G. Freund, and P. Willett. 1983. Automatic spelling correction using a trigram similarity measure. *Information Processing and Management*, 19(4):255–261.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case

study in part of speech tagging. *Computational Linguistics*, 21(4):543–565.

Evangelos Dermatas and George Kokkinakis. 1995. Automatic stochastic tagging of natural language texts. *Computational Linguistics*, 21(2):137–164.

Ted Dunning. 1994. Statistical identification of language. *Technical Report CRL MCCS 94-273i, New Mexico State University.*

Florian Gallwitz, Elmar Noeth, and Heinrich Niemann. 1996. A category based approach for recognition of out-of-vocabulary words. *Proceedings of International Conference on Spoken Language Processing*, 1:228–231.

Timothy J. Hazen and Bazzi Issam. 2001. A comparison and combination of methods for oov word detection and word confidence scoring. *Proceedings of ICASSP.*

IBM. 2001. Dictionary and linguistic tools. *http://booksrv1.raleigh.ibm.com/lingtool.*

Patrik Juola, Chris Hall, and Adam Boggs. 1994. Corpus based morphological segmentation by entropy changes. *Proceedings of the 3rd International Conference on the Cognitive Science of Natural Language Processing.*

Mitch Marcus and Marcinkiewicz B. Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330.

Andrei Mikheev. 1997. Automatic rule induction for unknown word guessing. *Computational Linguistics*, 23(3):405–423.

Youngja Park, Roy J. Byrd, and Branimir K. Boguraev. 2002. Automatic glossary extraction: Beyond terminology identification. *IBM Research Technical Report, RC22421.*

Joseph Pickett *et al.* 1996. The american heritage book of english usage: A practical and anthoritative guide to contemporary english. *Houghton Mifflin Company.*

Yael Ravin, Nina Wacholder, and Misook Choi. 1997. Disambiguation of proper names in text. *17th Annual ACM-SIGIR Conference.*

Claude E. Shannon. 1951. Prediction and entropy of printed english. *The Bell System Technical Journal*, January:50–64.

Larry Wall and Randal L. Schwartz. 1992. Programming perl. *O'Reilly & Associates, Inc.*

Ralph Weischedel, Marie Meeter, Richard Schwartz, Lance Ramshaw, and Jeff Palmucci. 1993. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, (2):359–382.