GIST-IT: Summarizing Email Using Linguistic Knowledge and Machine Learning

Evelyne Tzoukermann
Bell Labs, Lucent Technologies
700 Mountain Avenue
Murray Hill, NJ, 07974, USA
evelyne@lucent.com

Smaranda Muresan
Columbia University
500 W 120th Street
New York, NY, 10027, USA
smara@cs.columbia.edu

Judith L. Klavans
Columbia University
Center for Research on Information Access
535 W 114th Street
New York, NY, 10027, USA
klavans@cs.columbia.edu

Abstract

We present a system for the automatic extraction of salient information from email messages, thus providing the gist of their meaning. Dealing with email raises several challenges that we address in this paper: heterogeneous data in terms of length and topic. Our method combines shallow linguistic processing with machine learning to extract phrasal units that are representative of email content. The GIST-IT application is fully implemented and embedded in an active mailbox platform. Evaluation was performed over three machine learning paradigms.

Introduction

The volume of email messages is huge and growing. A qualitative and quantitative study of email overload [Whittaker and Sidner (1996)] shows that people receive a large number of email messages each day (~ 49) and that 21% of their inboxes (about 334 messages) are long messages (over 10 Kbytes). Therefore summarization techniques adequate for real-world applications are of great interest and need [Berger and Mittal (2000), McKeown and Radev (1995), Kupiec et al (1995), McKeown et al (1999), Hovy (2000)].

In this paper we present GIST-IT, an automatic email message summarizer that will convey to the user the gist of the document through topic phrase extraction, by combining linguistic and machine learning techniques.

Email messages and web documents raise several challenges to automatic text processing, and the summarization task addresses most of them: they are free-style text, not always syntactically or grammatically well-formed, domain and genre independent, of variable length and on multiple topics. Furthermore, due to the lack of well-formed syntactic and grammatical structures, the granularity of document extracts presents another level of complexity. In our work, we address the extraction problem at phrase-level [Ueda et al (2000), Wacholder et al (2000)], identifying salient information that is spread across multiple sentences and paragraphs.

Our novel approach first extracts simple noun phrases as candidate units for representing document meaning and then uses machine learning algorithms to select the most prominent ones. This combined method allows us to generate an informative, generic, "at-a-glance" summary.

In this paper, we show: (a) the efficiency of the linguistic approach for phrase extraction in comparing results with and without filtering techniques, (b) the usefulness of vector representation in determining proper features to identify contentful information, (c) the benefit of using a new measure of TF*IDF for the noun phrase and its constituents, (d) the power of machine learning systems in evaluating several classifiers in order to select the one performing the best for this task.

1 Related work

Traditionally a document summary is seen as a small, coherent prose that renders to the user the important meaning of the text. In this framework most of the research has focused on extractive summaries at sentence level. However, as discussed in [Boguraev and Kennedy (1999)], the meaning of 'summary' should be adjusted depending on the information management task for which it is used. Key phrases, for example, can be seen as semantic metadata that summarize and characterize documents [Witten et al (1999), Turney (1999)]. These approaches select a set of candidate phrases (sequence of one, two or three consecutive stemmed, non-stop words) and then apply machine learning techniques to classify them as key phrases or not. But dealing only with n-grams does not always provide good output in terms of a summary (see discussion in Section 5.4).

Wacholder (1998) proposes a linguistically-motivated method for the representation of the document aboutness: 'head clustering'. A list of simple noun phrases is first extracted, clustered by head and then ranked by the frequency of the head. Klavans et al (2000) report on the evaluation of 'usefulness' of head clustering in the context of browsing applications, in terms of quality and coverage.

Other researchers have used noun-phrases quite successfully for information retrieval task [Strzalkowski et al (1999), Sparck-Jones (1999)]. Strzalkowski et al (1999) uses head + modifier pairs as part of a larger system which constitutes the "stream model" that is used for information retrieval. They treat the head-modifier relationship as an "ordered relation between otherwise equal elements", emphasizing that for some tasks, the syntactic head of the NP is not necessarily a semantic head, and the modifier is not either necessarily a semantic modifier and that the opposite is often true. Using a machine learning approach, we proved this hypothesis for the task of gisting.

Berger and Mittal (2000) present a summarization system named OCELOT, based on probabilistic models, which provides the gist of web documents. Like email messages, web documents are also very heterogeneous and their unstructured nature pose equal difficulties.

In this paper, we propose a novel technique for summarization that combines the linguistic approach of extracting simple noun phrases as possible candidates for document extracts, and the use of machine learning algorithms to automatically select the most salient ones.

2 System architecture

The input to GIST-IT is a single email message. The architecture, presented in Figure 1 consists of four distinct functional components. The first module is an email preprocessor developed for Text-To-Speech

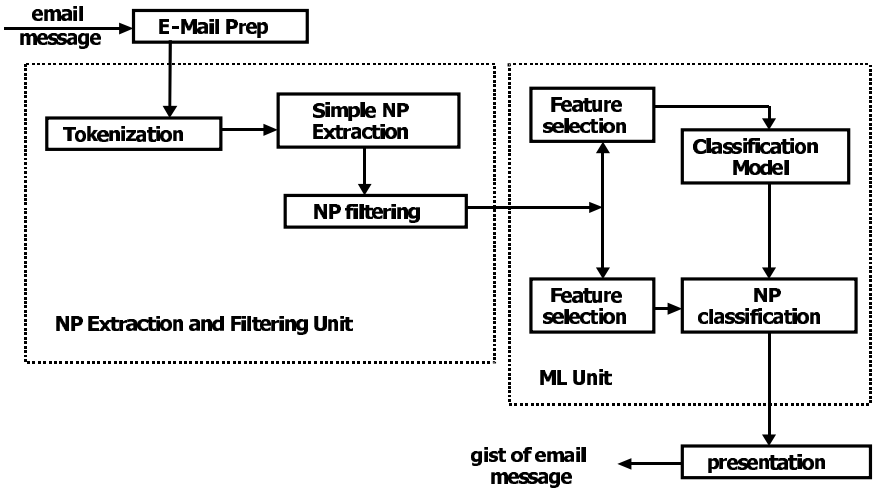

Figure 1 System Architecture

applications. The second component is a shallow text processing unit, which is actually a pipeline of modules for extraction and filtering of simple NP candidates. The third functional component is a machine learning unit, which consists of a feature selection module and a text classifier. This module uses a training set and a testing set that were devided from our email corpus. In order to test the performance of GIST-IT on the task of summarization, we use a heterogeneous collection of email messages in genre, length, and topic. We represent each email as a set of NP feature vectors. We used 2,500 NPs extracted from 51 email messages as a training set and 324 NPs from 8 messages for testing. Each NP was manually tagged for saliency by one of the authors and we are planning to add more judges in the future. The final module deals with presentation of the gisted email message.

2.1 The Email Preprocessor

This module uses finite-state transducer technology in order to identify message content. Information at the top of the message related to "From/To/Date" as well as the signature block are separated from the message content.

2.2 Candidate Simple Noun Phrase Extraction and Filtering Unit

This module performs shallow text processing for extraction and filtering of simple NP candidates, consisting of a pipeline of three modules: text tokenization, NP extraction, and NP filtering. Since the tool was created to preprocess email for speech output, some of the text tokenization suitable for speech is not accurate for text processing and some modifications needed to be implemented (e.g. email preprocessor splits acronyms like DLI2 into DLI 2). The noun phrase extraction module uses Brill's POS tagger [Brill (1992)]and a base NP chunker [Ramshaw and Marcus (1995)]. After analyzing some of these errors, we augmented the tagger lexicon from our training data and we added lexical and contextual rules to deal mainly with incorrect tagging of gerund endings. In order to improve the accuracy of classifiers we perform linguistic filtering, as discussed in detail in Section 3.1.2.

2.3 Machine Learning Unit

The first component of the ML unit is the feature selection module to compute NP vectors. In the training phase, a model for identifying salient simple NPs is created. The training data consist of a list of feature vectors already classified as salient/non-salient by the user. Thus we rely on user-relevance judgments to train the ML unit. In the extraction phase this unit will classify relevant NPs using the model generated during training. We applied three machine learning paradigms (decision trees, rule induction algorithms, and decision forest) evaluating three different classifiers.

2.4 Presentation

The presentation of the message gist is a complex user interface issue with its independent set of problems. Depending on the application and its use, one can think of different presentation techniques. The gist of the message could be the set of NPs or the set of sentences in which these NPs occur so that the added context would make it more understandable to the user. We do not address in this work the disfluency that could occur in listing a set of extracted sentences, since the aim is to deliver to the user the very content of the message even in a raw fashion. GIST-IT is to be used in an application where the output is synthesized speech. The focus of this paper is on extracting content with GIST-IT, although presentation is a topic for future research.

3 Combining Linguistic Knowledge and Machine Learning for Email Gisting

We combine symbolic machine learning and linguistic processing in order to extract the salient phrases of a document. Out of the large syntactic constituents of a sentence, e.g. noun phrases, verb phrases, and prepositional phrases, we assume that noun phrases (NPs) carry the most contentful information about the document, even if sometimes the verbs are important too, as reported in the work by [Klavans and Kan (1998)]. The problem is that no matter the size of a document, the number of informative noun phrases is very small comparing with the number of all noun

phrases, making selection a necessity. Indeed, in the context of gisting, generating and presenting the list of all noun phrases, even with adequate linguistic filtering, may be overwhelming. Thus, we define the extraction of important noun phrases as a classification task, applying machine learning techniques to determine which features associated with the candidate NPs classify them as salient vs. non-salient. We represent the document -- in this case an email message -- as a set of candidate NPs, each of them associated with a feature vector used in the classification model. We use a number of linguistic methods both in the extraction and in the filtering of candidate noun phrases, and in the selection of the features.

3.1 Candidate NPs

Noun phrases were extracted using Ramshaw and Marcus's base NP chunker [Ramshaw and Marcus (1995)]. The base NP is either a simple NP as defined by Wacholder (1998) or a conjunction of two simple NPs. Since the feature vectors used in the classifier scheme are simple NPs we used different heuristics to automatically split the conjoined NPs (CNP) into simple ones (SNP), properly assigning the premodifiers. Table 1 presents such an example:

CNP: *physics/NN and/CC biology/NN skilled/JJ researchers/NNS*
SNP1: *physics/NN skilled/JJ researchers/NNS*
SNP2: *biology/NN skilled/JJ researchers/NNS*

Table 1 Splitting Complex NPs into Simple NPs

3.1.2 Filtering simple NPs

Since not all simple noun phrases are equally important to reflect the document meaning, we use well-defined linguistic properties to extract only those NPs (or parts of NPs) that have a greater chance to render the salient information. By introducing this level of linguistic filtering before applying the learning scheme, we improve the accuracy of the classifiers, thus obtaining better results (see discussion in sections 4.1.3 and 5.3). We performed four filtering steps:

1. Inflectional morphological processing. English nouns have only two kinds of inflection: an affix that marks plural and an affix that marks possessive.

2. Removing unimportant modifiers. In this second step we remove the determiners that accompany the nouns and also the auxiliary words most and more that form the periphrastic forms of comparative and superlative adjectives modifying the nouns.

3. Remove common words. We used a list of 571 common words used in IR systems in order to further filter the list of candidate NPs. Thus, words like *even, following, every,* are eliminated from the noun phrase structure. (i.e. "even more detailed information" and "detailed information" will also be grouped together).

4. Remove 'empty' nouns. Words like *lot, group, set, bunch* are considered 'empty' nouns in the sense that they have no contribution to the noun phrase meaning. For example the meaning of the noun phrases like "group of students", "lots of students" or "bunch of students" is given by the noun "students". In order not to bias the extraction of empty nouns we used three different data collections: Brown corpus, Wall Street Journal, and a set of 4000 email messages (most of which were collected during a conference organization). Our algorithm was a simple one: we extracted all the nouns that appear in front of the preposition "of" and then sorted them by frequency of appearance in all three corpora and used a threshold to select the final list. We generated a set of 141 empty nouns that we used in this forth step of filtering process.

3.2 Feature Selection

We select a set of nine features that fall into three categories: linguistic, statistical (frequency-based) and positional. These features capture information about the relative importance of NPs to the document meaning.

Several studies rely on linguistic intuition that the head of the noun phrase makes a greater contribution to the semantics of the nominal group than the modifiers. For some NLP tasks, the head is not necessarily the most important item of the noun phrase. In analyzing email messages from the perspective of finding salient NPs, we claim

that the constituents of the NP have often as much semantic content as the head. This opinion is also supported in the work of [Strzalkowski et al (1999)]. In many cases, the meaning of the NP is given equally by modifier(s) -- usually nominal modifiers(s) -- and head. Consider the following list of simple NPs selected as candidates:

(1) "conference workshop announcement"
(2) "international conference"
(3) "workshop description"
(4) *"conference deadline"*

In the case of noun phrase (1) the importance of the noun phrase is found in the two noun modifiers: *conference* and *workshop* as much as in the head *announcement*. We test this empirical observation by introducing as a separate feature in the feature vector, a new TF*IDF measure that counts for both the modifiers and the head of the noun phrase, thus seeing the NP as a sequence of equally weighted elements. For the example above the new feature will be:

$TF*IDF_{conference} + TF*IDF_{workshop} + TF*IDF_{announcement}$

We divided the set of features into three groups: one associated with the head of the noun phrase, one associated with the whole NP and one that represents the new TF*IDF measure discussed above. Since we want to use this technique on other types of documents, all features are independent of the text type or genre. For example, in the initial selection of our attributes we introduced as separate features the presence or the absence of NPs in the subject line of the email and in the headline of the body. Kilander (1996) pointed out that users estimate that "subject lines can be useful, but also devastating if their importance is overly emphasized". Based on this study and also on our goal to provide a method that is domain and genre independent we decided not to consider the subject line and the headlines as separate features, but rather as weights included in the TF*IDF measures as presented below. Another motivation for this decision is that in email processing the correct identification of headlines is not always clear.

3.2.1 Features associated with the Head

We choose two features to characterize the head of the noun phrases:

head_tfidf – the TF*IDF measure of the head of the candidate NP.

head_focc - The first occurrence of the head in text (the numbers of words that precede the head divided by the total number of words in the document).

3.2.2 Features associated with the whole NP

We select six features that we consider relevant in association with the whole NP:

np_tfidf – the TF*IDF measure associated with the whole NP.

np_focc - The first occurrence of the noun phrase in the document.

np_length_words - Noun phrase length measured in number of words, normalized by dividing it with the total numbers of words in the candidate NPs list.

np_length_chars - Noun phrase length measured in number of characters, normalized by dividing it with the total numbers of characters in the candidate NPs list.

sent_pos - Position of the noun phrase in sentence: the number of words that precede the noun phrase, divided by the sentence length. For noun phrases in the subject line and headlines (which are usually short and will be affected by this measure), we consider the maximum length of sentence in document as the normalization factor.

par_pos - Position of noun phrase in paragraph, same as sent_pos, but at the paragraph level.

3.2.3 Feature that considers all constituents of the NP equally weighted

m_htfidf - the new TF*IDF measure that take into consideration the importance of the modifiers.

In computing the TF*IDF measures (*head_tfidf, np_tfidf, m_tfidf*), weights w_i, were assigned to account for the presence in the subject line and/or headline.

w_{i1} – if the head appears both in the subject line and headline;

w_{i2} – if the head appears only in the subject line;

w_{i3} – if the head appears only in headlines where $w_{i1} > w_{i2} > w_{i3}$.

These weights were manually chosen after a set of experiments, but we plan to use either

a regression method or explore with genetic algorithms to automatically learn them.

3.3 Three Paradigms of Supervised Machine Learning

Symbolic machine learning is used in conjunction with many NLP applications (syntactic and semantic parsing, POS tagging, text categorization, word sense disambiguation). In this paper we compare three symbolic learning techniques applied to the task of salient NP extraction: decision tree, rule induction learning and decision forests.

We tested the performance of an axis-parallel decision tree, C4.5 [Quinlan (1993)]; a rule learning system RIPPER [Cohen (1995)] and a decision forest classifier (DFC) [Ho (1998)]. RIPPER allows the user to specify the loss ratio, which indicates the ratio of the cost of a false positive to the cost of a false negative, thus allowing the trade off between precision and recall. This is crucial for our analysis since we deal with sparse data set (in a document the number of salient NPs is much smaller than the number of irrelevant NPs). Finally we tried to prove that a combination of classifiers might improve accuracy, increasing both precision and recall. The Decision Forest Classifier (DFC) uses an algorithm for systematically constructing decision trees by pseudo-randomly selecting subsets of components of feature vectors. It implements different splitting functions. In the setting of our evaluation we tested the information gain ratio (similar to the one used by Quinlan in C4.5). An augmented feature vector (pairwise sums, differences, and products of features) was used for this classifier.

4 Evaluation and Experimental Results

Since there are many different summaries for each document, evaluating summaries is a difficult problem. Extracting the salient noun phrases is the first key step in the summarization method that we adopt in this paper. Thus, we focus on evaluating the performance of GIST-IT on this task, using three classification schemes and two different feature settings.

4.1 Evaluation Scheme

There are several questions that we address in this paper:

4.1.1 What features or combination of features are important in determining the degree of salience of an NP?

Following our assumption that each constituent of the noun phrase is equally meaningful, we evaluate the impact of adding *m_htfidf* (see section 3.2.3), as an additional feature in the feature vector. This is shown in Table 2 in the different feature vectors *fv1* and *fv2*.

fv1- head_focc head_tfidf np_focc np_tfidf
np_length_words np_length_chars par_pos sent_pos
fv2 - head_focc head_tfidf **m_htfidf** np_focc np_tfidf
np_length_words np_length_chars par_pos sent_pos

Table 2 Two feature settings to evaluate the impact of m_htfidf

4.1.2 What classification scheme is more adequate to our task?

We evaluate the performance of three different classifiers in the task of extracting salient noun phrases. As measures of performance we use precision (p) and recall (r). The evaluation was performed according to what degree the output of the classifiers corresponds to the user judgments.

Feature vectors	C4.5		Ripper		DFC	
	p	r	p	r	p	r
fv1	73.3	78.6	83.6	71.4	80.3	83.5
fv2	70	88.9	85.7	78.8	85.7	87.9

Table 3 Evaluation of two feature vectors using three classifiers

Table 3 shows our results that answer these two questions. The table rows represent the two feature vectors we are comparing, and the columns correspond to the three classifiers chosen for the evaluation.

4.1.3 Is linguistic filtering an important step in extracting salient NPs?

In the third evaluation we analyse the impact of linguistic filtering on the classifier's performance. It turns out that results show major improvements, from 69.2% to 85.7% for precision of fv2, and from 56.25% to 87.9% for recall of fv2. For detailed results, see [Muresan et al, (2001)].

4.1.4 After the filtering and classification, are noun phrases good candidates for representing the gist of an email message?

In order to answer this question, we compare the output of GIST-IT on one email with the results of KEA system [Witten et al (1999)] that uses a 'bag-of-words' approach to key phrase extraction (see Table 4).

module	*Perl module wordne*
sort of batch	*interface*
WordNet data	*'wn' command line program*
accesses	*simple easy perl interface*
the WordNet	*included man page*
lots of WordNet	*wordnet*
WordNet perl	*wordnet.pm module*
QueryData	*wordnet system*
wn	*wordnet package*
perl module	*query perl module*
extracting	*command line*
use this module	*wordnet relation*
extracting lots	*wordnet data*
WordNet system	*free software*
www.cogsci.princeton.edu	*querydata*

Table 4 KEA (left) vs GIST-IT output (right)

5 Discussion of results

The results shown indicate that best system performance reached 87.9% recall and 85.7% precision. Although these results are very high, judging NP relevance is a complex and highly variable task. In the future, we will extend the gold standard with more judges, more data, and thus a more precise standard for measurement.

5.1 The right selection of features

Feature selection has a decisive impact on overall performance. As seen in Table 2, *fv2* has *m_htfidf* as an additional feature, and its performance shown in Table 3 is superior to fv1; the DFC classifier shows an increase both in precision and recall. These results support the original hypothesis that in the context of gisting, the syntactic head of the noun phrase is not always the semantic head, and modifiers can also have an important role.

5.2 Different classification models

The effectiveness of different classification schemes in the context of our task is discussed here. As shown in Table 3, C4.5 performs well especially in terms of recall. RIPPER, as discussed in [Cohen (1995)], is more appropriate for noisy and sparse data collection than C4.5, showing an improvement in precision. Finally, DFC which is a combination of classifiers, shows improved performance. The classifier was run with an augumented feature vector that included pairwise sums, differences and products of the features.

5.3 Impact of linguistic knowledge

As shown in previous section, DFC performed best in our task, so we chose only this classifier to present the impact of linguistic knowledge. Linguistic filtering improved precision and recall, having an important role especially on *fv2*, where the new feature *m_tfidf* was used. This is explained by the fact that the filtering presented in section 3.1.2 removed the noise introduced by unimportant modifiers, common and empty nouns, thus giving this new feature a larger impact.

5.4 Noun phrases are better than n-grams

Presenting the gist of an email message by phrase extraction addresses one obvious question: can any phrasal extract represent the content of a document, or must a well defined linguistic phrasal structure be used? To answer this question we compare the results of our system that extract linguistically principled phrasal units, with KEA output, that extracts bigrams and trigrams as key phrases [Witten et al (1999)]. Table 4 shows the results of the KEA system. Due to the n-gram approach, KEA output contains phrases like *sort of batch*, *extracting lots*, *wn*, and even urls that are unlikely to represent the gist of a document.

Conclusion and future work

In this paper we presented a novel technique for document gisting suitable for domain and genre independent collections such as email messages. The method extracts simple noun phrases using linguistic techniques and then use machine learning to classify them as salient for the document content. We evaluated the system in different experimental settings using three classification models. In analyzing the structure of NPs, we demonstrated that the modifiers of a noun phrase can be semantically as important as the head for the

task of gisting. GIST-IT is fully implemented, evaluated, and embedded in an application, which allows user to access a set of information including email, finances, etc.

We plan to extend our work by taking advantage of structured email, by classifying messages into folders, and then by applying information extraction techniques. Since NPs and machine learning techniques are domain and genre independent, we plan to test GIST-IT on different data collections (e.g. web pages), and for other knowledge management tasks, such as document indexing or query refinement. Additionally, we plan to test the significance of the output for the user, i.e. whether the system provide informative content and adequate gist of the message.

References

Berger, A.L and Mittal, V.O (2000). *OCELOT:A system for summarizing web pages*. In Proceedings of the 23rd Annual International ACM SIGIR, Athens, Greece, pp 144-151.

Brill, E. (1992). *A Simple Rule-based Part of Speech Tagger*. In Proceedings of the Third Conference on ANLP. Trento, Italy; 1992

Boguraev, B. and Kennedy, C. (1999). *Salience-based content characterisation of text documents*. In I. Mani and T. Maybury, M., editors, Advances in Automatic Text Summarization, pp 99-111. The MIT Press.

Cohen, W. (1995). *Fast Effective Rule Induction*. Machine-Learning: Proceedings of the Twelfth International Conference.

Ho, T.K (1998). *The random subspace method for constructing decision forests*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8).

Hovy, E.H (2000). *Automated Text Summarization*. In R. Mitkov, editor, Oxford University Handbook of Computational Linguistics. Oxford Univ. Press.

Kilander, F. (1996). *Properties of electronic texts for classification purposes as suggested by users*.

Klavans, J.L., Wacholder, N. and Evans, D.K. (2000) *Evaluation of computational linguistic techniques for identifying significant topics for browsing applications*. In Proceedings (LREC-2000), Athens. Greece.

Klavans, J.L. and Kan, M-Y. (1998).*Role of verbs in document analysis*. In proceedings of COLING/ACL 98.

Kupiec, J., Pedersen, J. and Chen, F. (1995). *A trainable document summarizer*. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 68-73, Seattle, WA.

McKeown, K.R, Klavans, J.L, Hatzivassiloglou, V., Barzilay, R. and Eskin, E. (1999). *Towards multidocument summarization by reformulation: Progress and prospects*. In Proceedings of AAAI'99.

McKeown, K.R and Radev, D.R (1995). *Generating summaries of multiple news articles*. In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 74-82, Seattle, WA.

Muresan, S., Tzoukermann, E. and Klavans, J.L. (2001). *Email Summarization Using Linguistic and Machine Learning Techniques*. In Proceedings of CoNLL 2001 ACL Workshop, Toulouse, France.

Murthy, S.K., Kasif, S., Salzberg, S. and Beigel, R. (1993). *OC1: Randomized Induction of Oblique Decision Trees*. Proceedings of the Eleventh National Conference on Artificial Intelligence, pp. 322--327, Washington, D.C.

Quinlan, J.R (1993). *C4.5: Program for Machine Learning*. Morgan Kaufmann.

Ramshaw, L.A. and Marcus, M.P. (1995). *Text Chunking Using Transformation-Based Learning*. In Proceedings of Third ACL Workshop on Very Large Corpora, MIT.

Sparck-Jones, K. (1999). *What Is The Role of NLP in Text Retrieval*. In T. Strzalkowski, editor, Natural Language Information Retrieval. Kluwer, Boston, MA.

Strzalkowski, T., Lin, F., Wang, J., and Perez-Carballo, J. (1999). Evaluating natural language processing techniques for information retrieval. In T. Strzalkowski, editor, Natural Language Information Retrieval. Kluwer, Boston, MA.

Turney, P.D. (2000). *Learning algorithms for keyphrase exraction*. Information Retrieval, 2(4): pp 303-336.

Ueda, Y., Oka M., Koyama T. and Miyauchi T (2000). *Toward the "at-a-glance" summary: Phrase-representation summarization method*. In Proceedings of COLING 2000.

Wacholder, N. (1998). *Simplex NPS sorted by head: a method for identifying significant topics within a document*, In Proceedings of the COLING-ACL Workshop on the Computational Treatment of Nominals.

Whittaker, S. and Sidner, C. *Email overload: Exploring personal information management of email*. In Proceedings of CHI'96. p. 276-283. NY:ACM Press

Witten, I.H, Paynter, G.W., Frank E., Gutwin C. and Nevill-Manning, C.G (1999). *KEA: Practical automatic keyphrase extraction*. In Proceedings of DL'99, pp 254-256.