Combining Linguistic and Machine Learning Techniques for Email Summarization

Smaranda Muresan
Dept. of Computer Science
Columbia University
500 West 120 Street
New York, NY, 10027
smara@cs.columbia.edu

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

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

Abstract

This paper shows that linguistic techniques along with machine learning can extract high quality noun phrases for the purpose of providing the gist or summary of email messages. We describe a set of comparative experiments using several machine learning algorithms for the task of salient noun phrase extraction. Three main conclusions can be drawn from this study: (i) the modifiers of a noun phrase can be semantically as important as the head, for the task of gisting, (ii) linguistic filtering improves the performance of machine learning algorithms, (iii) a combination of classifiers improves accuracy.

1 Introduction

In this paper we present a comparative study of symbolic machine learning models applied to natural language task of summarizing email messages through topic phrase extraction.

Email messages are domain-general text, they are unstructured and not always syntactically well formed. These characteristics raise challenges for automatic text processing, especially for the summarization task. Our approach to email summarization, implemented in the GIST-IT system, is to identify topic phrases, by first extracting noun phrases as candidate units for representing document meaning and then using machine learning algorithms to select the most salient ones.

The comparative evaluation of several machine learning models in the settings of our experiments indicates that : (i) for the task of gisting the modifiers of the noun phrase are equally as important as the head, (ii) noun phrases are better than n-grams for the phrase-level representation of the document, (iii) linguistic filtering enhances machine learning techniques, (iv) a combination of classifiers improves accuracy.

Section 2 of the paper outlines the machine learning aspect of extracting salient noun phrases, emphasizing the features used for classification and the symbolic machine learning models used in the comparative experiments. Section 3 presents the linguistic filtering steps that improve the accuracy of the machine learning algorithms. Section 4 discusses in detail our conclusions stated above.

2 Machine Learning for Content Extraction

Symbolic machine learning has been applied successfully in conjunction with many NLP applications (syntactic and semantic parsing, POS tagging, text categorization, word sense disambiguation) as reviewed by Mooney and Cardie (1999). We used machine learning techniques for finding salient noun phrases that can represent the summary of an email message. This section describes the three steps involved in this classification task: 1) what representation is appropriate for the information to be classified as relevant or non-relevant (candidate phrases), 2) which features should be associated with each candidate, 3) which classification models should be used.

Case 1	
CNP:	scientific/JJ and/CC technical/JJ articles/NNS
SNP1:	scientific/JJ articles/NNS
SNP2:	technical/JJ articles/NNS
Case 2	
CNP:	scientific/JJ thesauri/NNS and databases/NNS
SNP1:	scientific/JJ thesauri/NNS
SNP2:	scientific/JJ databases/NNS
Case 3	
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: Resolving Coordination of NPs

2.1 Candidate Phrases

Of the major 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, a well-supported hypothesis (Smeaton, 1999; Wacholder, 1998).

As considered by Wacholder (1998), the simple NPs are the maximal NPs that contain premodifiers but not post-nominal constituents such as prepositions or clauses. We chose simple NPs for content representation because they are semantically and syntactically coherent and they are less ambiguous than complex NPs. For extracting simple noun phrases we first used Ramshaw and Marcus's base NP chunker (Ramshaw and Marcus, 1995). The base NP is either a simple NP or a coordination of simple NPs. We used heuristics based on POS tags to automatically split the coordinate NPs into simple ones, properly assigning the premodifiers. Table 1 presents some coordinate NPs (CNP) encountered in our data collection and the results of our algorithm which split them into simple NPs (SNP1 and SNP2).

2.2 Features used for Classification

The choice of features used to represent the candidate phrases has a strong impact on the accuracy of the classifiers (e.g. the number of examples needed to obtain a given accuracy on the test data, the cost of classification). For our classification task of determining if a noun phrase is salient or not to the document meaning, we chose a set of nine features.

Several studies rely on the linguistic intuition that the head of the noun phrase makes a greater contribution to the semantics of the nominal group than the modifiers. However, for some specific tasks in NLP , the head is not necessarily the most semantically important part of the noun phrase. In analyzing email messages from the perspective of finding salient NPs, we claim that the modifier(s) of the noun phrase - usually nominal modifiers(s), often have as much semantic content as the head. This opinion is also supported in the work of Strzalkowski et al. (1999), where syntactic NPs are captured for the goal of extracting their semantic content but are processed as an "ordered" string of words rather than a syntactic unit. Thus we introduce as a separate feature in the feature vector, a new TF*IDF measure which consider the NP as a sequence of equally weighted elements, counting individually the modifier(s) and the 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 the first noun phrase, for example, its importance is found in the two noun modifiers: *conference* and *workshop* as much as in the head *announcement*, due to their presence as heads or modifiers in the candidate NPs 2-4. Our new feature will be: $TF * IDF_{conference} + TF * IDF_{workshop} + TF * IDF_{announcement}$. Giving these linguistic observations we divided the set of features into three groups, as we mentioned also in (Tzoukermann et al., 2001): 1) one associated with the head of the noun phrase; 2) one associated with the whole NP and 3) one that represents the new TF*IDF measure discussed above.

2.2.1 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. For the NP in example (1) this feature will be $TF * IDF_{announcement}$.
- **head_focc**: The position of the first occurrence of the head in text (the number of words that precede the first occurrence of the head divided by the total number of words in the document).

2.2.2 Features associated with the whole NP

We select six features that we consider relevant in determining the relative importance of the noun phrase:

- **np_tfidf**: the TF*IDF measure of the whole NP. For the NP in the example (1) this feature will be $TF * IDF_{conference\ workshop\ announcement}$.
- **np_focc**: The position of 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 number of words in the candidate NP list.
- **np_length_chars**: Noun phrase length measured in number of characters, normalized by dividing it with the total number of characters in the candidate NPs list.
- **sent_pos**: Position of the noun phrase in the sentence: the number of words that precede the noun phrase, divided by sentence length. For noun phrases in the subject line (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.

2.2.3 Feature that considers all constituents of the NP equally weighted

One of the important hypotheses we tested in this work is that both the modifiers and the head of NP contribute equally to its salience. Thus we consider **mh_tfidf** as an additional feature in the feature vector.

- **mh_tfidf**: the new TF*IDF measure that takes also into consideration the importance of the modifiers. In our example the value of this feature will be : $TF * IDF_{conference} + TF * IDF_{workshop} + TF * IDF_{announcement}$

In computing the TF*IDF measures (*head_tfidf*, *np_tfidf*, *mh_tfidf*), specific weights, w_i, were assigned to account for the presence in the email subject line and/or headlines in the email body.

- w_{i1}: presence in the subject line and headline
- w_{i2}: presence in the subject line
- w_{i3}: presence in headlines where $w_{i1} > w_{i2} > w_{i3}$.

These weights were manually chosen after a set of experiments, but we plan to use a regression method to automatically learn them.

2.3 Symbolic Machine Learning Models

We compared three symbolic machine learning paradigms (decision trees, rule induction and decision forests) applied to the task of salient NP extraction, evaluating five classifiers.

2.3.1 Decision Tree Classifiers

Decision trees classify instances represented as feature vectors, where internal nodes of the tree test one or several attributes of the instance and where the leaves represent categories. Depending on how the test is performed at each node, there exists two types of decision tree classifiers: axis parallel and oblique. The axis-parallel decision trees check at each node the value of a single attribute. If the attributes are numeric, the test has the form $x_i > t$, where x_i is one of the attribute of an instance and *t* is the threshold. Oblique decision trees test a linear combination of attributes at each internal node:

$$\sum_{i=1}^{n} a_i x_i + a_{n+1} > 0$$

where $a_i, ..., a_{n+1}$ are real-valued coefficients. We compared the performance of C4.5, an axis-parallel decision tree classifier (Quinlan, 1993) and OC1, an oblique decision tree classifier (Murthy et al., 1993).

2.3.2 Rule Induction Classifiers

In rule induction, the goal is to learn the smallest set of rules that capture all the generalisable knowledge within the data. Rule induction classification is based on firing rules on a new instance, triggered by the matching feature values to the left-hand side of the rules. Rules can be of various normal forms and can be ordered. However, the appropriate ordering can be hard to find and the key point of many rule induction algorithms is to minimize the search strategy through the space of possible rule sets and orderings. For our task, we test the effectiveness of two rule induction algorithms : C4.5rules that form production rules from unpruned decision tree, and a fast top-down propositional rule learning system, RIPPER (Cohen, 1995). Both algorithms first construct an initial model and then iteratively improve it. C4.5rules improvement strategy is a greedy search, thus potentially missing the best rule set. Furthermore, as discussed in (Cohen, 1995), for large noisy datasets RIPPER starts with an initial model of small size, while C4.5rules starts with an over-large initial model. This means that RIPPER's search is more efficient for noisy datasets and thus is more appropriate for our data collection. It also allows the user to specify the loss ratio, which indicates the ratio of the cost of false positives to the cost of false negatives, thus allowing a trade off between precision and recall. This is crucial for our analysis since we deal with sparse data due to the fact that in a document the number of salient NPs is much smaller than the number of irrelevant NPs.

2.3.3 Decision Forest Classifier

Decision forests are a collection of decision trees together with a combination function. We test the performance of DFC (Ho, 1998), a decision forest classifier that systematically constructs decision trees by pseudo-randomly selecting subsets of components of feature vectors. The advantage of this classifier is that it combines a set of different classifiers in order to improve accuracy. 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.

3 Linguistic Knowledge Enhances Machine Learning

Not all simple noun phrases are equally important to reflect document meaning. Boguraev and Kennedy (1999) discuss the issue that for the task of document gisting, topical noun phrases are usually noun-noun compounds. In our work, we rely on ML techniques to decide which are the salient NPs, but we claim that a shallow linguistic filtering applied before the learning process improves the accuracy of the classifiers. We performed four filtering steps:

1. **Inflectional morphological processing:** Grouping inflectional variants together can help especially in case of short documents (which is sometimes the case for email messages). English nouns have only two kinds of regular inflection: a suffix for the plural mark and another suffix for the possessive one.

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 (e.g. "the most complex morphology" will be filtered to "complex morphology").

3. **Removing 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.

4. **Removing empty nouns:** Words like *lot*, *group*, *set*, *bunch* are considered empty heads. For example the primary concept of the noun phrases like "group of students", "lots of students" or "bunch of students" is given by the noun "students". We extracted all the nouns that appear in front of the preposition "of" and then sorted them by frequency of appearance. A threshold was then used to select the final list (Klavans et al., 1990). Three different data collections were used: the Brown corpus, the Wall Street Journal, and a set of 4000 email messages (most of them related to a conference organization). We generated a set of 141 empty nouns that we used in this forth step of the filtering process.

4 Results and Discussion

One important step in summarization is the discovery of the relevant information from the source text. Our approach was to extract the salient NPs using linguistic knowledge and machine learning techniques. Our evaluation corpus consists of a collection of email messages which is heterogeneous in genre, length, and topic. 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 human judge. We are planning to add more judges in the future and measure the interuser agreement.

This section outlines a comparative evaluation of five classifiers using two feature settings on the task of extracting salient NPs from email messages. The evaluation shows the following important results:

Result 1. In the context of gisting, the head-modifier relationship is an ordered relation between semantically equal elements.

We evaluate the impact of adding *mh_tfidf* (see section 2.2), as an additional feature in the feature vector. This is shown in Table 2 in the different feature vectors *fv1* and *fv2*. The first feature vector, *fv1*, contains the features in sections 2.2.1 and 2.2.2, while *fv2* includes as an additional feature *mh_tfidf*.

As can be seen from Table 3, the results of evaluating these two feature settings using five different classifiers, show that *fv2* performed better than *fv1*. For example, the DFC classifier shows an increase both in precision and recall. This allows us to claim that in the context of gisting, the syntactic head of the noun phrase is not always the semantic head, and modifiers can have also an important role.

One advantage of the rule-induction algorithms is that their output is easily interpretable by humans. Analyzing C4.5rules output, we gain an insight on the features that contribute most in the classification process. In case of *fv1*, the most important features are: the first appearance of the NP and its head (*np_focc*, *head_focc*), the length of NP in number of words (*np_length_words*) and the tf*idf measure of the whole NP and its head (*np_tfidf*, *head_tfidf*). For example:

- IF head_focc <= 0.0262172 AND np_tfidf > 0.0435465 THEN Relevant
- IF np_focc <= 0.912409 AND np_length_words > 0.0242424 THEN Relevant
- IF head_tfidf <= 0.0243452 AND np_tfidf <= 0.0435465 AND np_length_words <= 0.0242424 then Not_relevant

In case of *fv2*, the new feature *m_tfidf* impacts the rules for both Relevant and Not_relevant categories. It supercedes the need for *np_tfidf* and *head_tfidf*, as can be seen also from the rules below:

- IF mh_tfidf > 0.0502262 AND np_focc <= 0.892585 THEN Relevant
- IF mh_tfidf > 0.0180134 AND np_length_words > 0.0260708 THEN Relevant
- IF mh_tfidf <= 0.0223546 AND np_length_words <= 0.0260708 THEN Not_relevant
- IF mh_tfidf <= 0.191205 AND np_focc > 0.892585 THEN Not_relevant

Feature vector 1 (fv1)
head_focc head_tfidf np_focc np_tfidf np_length_chars np_length_words par_pos sent_pos
Feature vector 2 (fv2)
head_focc head_tfidf **mh_tfidf** np_focc np_tfidf np_length_chars np_length_words par_pos sent_pos

Table 2: Two feature settings to evaluate the impact of **mh_tfidf**

	C4.5		**OC1**		**C4.5 rules**		**Ripper**		**DFC**	
	p	r	p	r	p	r	p	r	p	r
fv1	73.3%	78.6%	73.7%	93%	73.7%	88.5%	83.6%	71.4%	80.3%	83.5%
fv2	70%	88.9%	82.3%	88%	73.7%	95%	85.7%	78.8%	85.7%	87.9%

Table 3: Evaluation of two feature vectors using five classifiers

Result 2. Classifiers' performance depends on the characteristics of the corpus, and combining classifiers improves accuracy

This result was postulated by evaluating the performance of five different classifiers in the task of extracting salient noun phrases. As measures of performance we use precision and recall . The evaluation was performed according to what degree the output of the classifiers corresponds to the user judgments and the results are presented in Table 3.

We first compare two decision tree classifiers: one which uses as the splitting function only a single feature (C4.5) and the other, the oblique tree classifier (OC1) which at each internal node tests a linear combination of features. Table 3 shows that OC1 outperforms C4.5.

Columns 4 and 5 from Table 3 show the relative performance of RIPPER and C4.5rules. As discussed in (Cohen, 1995), RIPPER is more appropriate for noisy and sparse data collection than C4.5rules. Table 3 shows that RIPPER performs better than C4.5rules in terms of precision.

Finally, we investigate whether a combination of classifiers will improve performance. Thus we choose the Decision Forest Classifier, DFC, to perform our test. DFC obtains the best results, as can be seen from column 6 of Table 3.

Result 3. Linguistic filtering is an important step in extracting salient NPs

As seen from Result 2, the DFC performed best in our task, so we chose only this classifier to present the impact of linguistic filtering. Table 4 shows that linguistic filtering improves precision and recall, having an important role especially on *fv2*, where the new feature, *mh_tfidf* was used (from 69.2% precision and 56.25% recall to 85.7% precision and 87.9% recall).

	without filtering		**with filtering**	
	precision	recall	precision	recall
fv1	75%	75%	80.3%	83.5%
fv2	69.2%	56.25%	85.7%	87.9%

Table 4: Evaluation of linguistic filtering

This is explained by the fact that the filtering presented in section 3 removed the noise introduced by unimportant modifiers, common and empty nouns.

Result 4. Noun phrases are better candidates than n-grams

Presenting the gist of an email message by phrase extraction addresses one obvious question: are noun-phrases better than n-grams for representing the document content? To answer this question we compared the results of our system, GIST-IT, that extracts linguistically well motivated phrasal units, with KEA output, that extracts bigrams and trigrams as key phrases using a Naïve Bayes model (Witten et al., 1999). Table 5 shows the results on one email message. The n-gram approach of KEA system extracts phrases like *sort of batch*, *extracting lots*, *wn*, and even URLs that are unlikely to represent the gist of a document. This is an indication that the linguistically motivated GIST-IT phrases are more useful for document gisting. In future work we will perform also a task-based evaluation of these two

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

Table 5: Salient phrase extraction with GIST-IT vs. KEA on one email message

approaches, to test usability.

5 Related Work

Machine learning has been successfully applied to different natural language tasks, including text summarization. A document summary is seen as a succinct and coherent prose that captures the meaning of the text. Prior work in document summarization has been mostly based on sentence extraction. Kupiec et al. (1995) use machine learning for extracting the most important sentences of the document. But extractive summarization relies on the properties of source text that emails typically do not have: coherence, grammaticality, well defined structure. Berger and Mittal (2000) present a summarization system, named OCELOT that provides the gist of the web documents based on probabilistic models. Their approach is closed related with statistical machine translation.

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, 2000). These approaches select a set of candidate phrases (bigrams or trigrams) and then apply Naïve Bayes learning 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. In (Boguraev and Kennedy, 1999) the "gist" of a document is seen as a sequence of salient objects, usually topical noun phrases, presented in a highlighted context. Their approach is similar to extracting technical terms (Justeson and Katz, 1995). Noun phrases are used also in IR task (Strzalkowski et al., 1999; Smeaton, 1999; Sparck Jones, 1999). The work of Strzalkowski et al. (1999) supports our hypothesis that for some NLP tasks (gisting, IR) the head+modifier relation of a noun phrase is in fact an ordered relation between semantically equally important elements.

6 Conclusions 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 uses machine learning to classify them as salient for the document content. The contributions of this work are:

1. From a linguistic standpoint, we demonstrated that the modifiers of a noun phrase can be as semantically important as the head for the task of gisting.

2. From a machine learning standpoint, we evaluated the power and limitation of sev-

eral classifiers: decision trees, rule induction, and decision forests classifiers.

3. We proved that linguistic knowledge can enhance machine learning by evaluating the impact of linguistic filtering before applying the learning scheme.

The study, the evaluation, and the results provide experimental grounds for research not only in summarization, but also in information extraction and topic detection.

References

A.L. Berger and V.O. Mittal. 2000. OCELOT:A system for summarizing web pages. In *Proceedings of the 23rd Anual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 144–151, Athens, Greece.

B. Boguraev and C. Kennedy. 1999. Salience-based content characterisation of text documents. In Interjit Mani and T. Maybury, Mark, editors, *Advances in Automatic Text Summarization*, pages 99–111. The MIT Press.

W. Cohen. 1995. Fast effective rule induction. In *Machine-Learning: Proceedings of the Twelfth International Conference*.

T.K. Ho. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

J. Justeson and S. Katz. 1995. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, (1):9–27.

J.L. Klavans, M.S. Chodorow, and N. Wacholder. 1990. From dictionary to knowledge base via taxonomy. In *Proceedings of the Sixth Conference of the University of Waterloo Centre for the New Oxford English Dictionary and Text Research: Electronic Text Research*, University of Waterloo, Canada.

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

R.J Mooney and C. Cardie. 1999. Symbolic machine learning for natural language processing. In *ACL'99 Tutorial*.

S.K. Murthy, S. Kasif, S. Salzberg, and R. Beigel. 1993. OC1: Randomized induction of oblique decision trees. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 322–327, Washington, D.C.

J.R Quinlan. 1993. *C4.5: Program for Machine Learning*. Morgan Kaufmann Publisher, San Mateo, California.

L.A. Ramshaw and M.P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of Third ACL Workshop on Very Large Corpora*.

A. Smeaton. 1999. Using NLP or NLP resources for information retrieval tasks. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*. Kluwer, Boston, MA.

K. Sparck Jones. 1999. What is the role for NLP in text retrieval. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*, pages 1–12. Kluwer, Boston, MA.

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

P.D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, May.

E Tzoukermann, S Muresan, and J.L. Klavans. 2001. GIST-IT: Summarizing email using linguistic knowledge and machine learning. In *Proceeding of the HLT and KM Workshop, EACL/ACL 2001*.

N. Wacholder. 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*, Montreal, Canada.

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