UNIVERSITÀ
DEGLI STUDI
DI UDINE

Università degli studi di Udine

Evaluating anaphora and coreference resolution to improve automatic keyphrase extraction

Original

Availability:
This version is available http://hdl.handle.net/11390/1123620 since 2018-02-12T16:21:08Z

Publisher:

Published
DOI:

Terms of use:
The institutional repository of the University of Udine (http://air.uniud.it) is provided by ARIC services. The aim is to enable open access to all the world.

Publisher copyright

(Article begins on next page)

19 April 2024

Evaluating anaphora and coreference resolution to improve automatic keyphrase extraction

Marco Basaldella, Giorgia Chiaradia and **Carlo Tasso**
{basaldella.marco.1, chiaradia.giorgia}@spes.uniud.it
carlo.tasso@uniud.it
Artificial Intelligence Laboratory
Università degli Studi di Udine
Via delle Scienze 208, Udine, Italy

Abstract

In this paper we analyze the effectiveness of using linguistic knowledge from coreference and anaphora resolution for improving the performance for supervised keyphrase extraction. In order to verify the impact of these features, we define a baseline keyphrase extraction system and evaluate its performance on a standard dataset using different machine learning algorithms. Then, we consider new sets of features by adding combinations of the linguistic features we propose and we evaluate the new performance of the system. We also use anaphora and coreference resolution to transform the documents, trying to simulate the cohesion process performed by the human mind. We found that our approach has a slightly positive impact on the performance of automatic keyphrase extraction, in particular when considering the ranking of the results.

1 Introduction

Automatic Keyphrase Extraction (henceforth AKE), i.e. the task of extracting a list of phrases of one or more words "that capture the main topics discussed in a given document" (Turney, 2000) is a natural language processing (herein NLP) task which received widespread attention in the last years, with applications, e.g., in the fields of digital libraries (Gutwin et al., 1999) or community modelling (De Nart et al., 2015).

Many AKE algorithms have been developed, which can be roughly divided into two categories (Hasan and Ng, 2014):

- *Supervised algorithms*: after the generation of candidate keyphrases (henceforth KPs) by means of linguistic knowledge, these candidates are associated to a set of *features* such as TF-IDF, position in the text, and so on; then, a supervised machine learning (herein ML) algorithm learns over a training set how to decide if a candidate is a suitable KP or not.

- *Unsupervised algorithms*: for example, the document is represented using a graph structure, whose nodes are candidate KPs. Then, the *popularity* of each candidate is evaluated using graph algorithms usually derived from the PageRank algorithm (Mihalcea and Tarau, 2004; Wan and Xiao, 2008). Other approaches include for example clustering-based algorithms, such as the one presented in (Liu et al., 2009), or techniques which rely on building a statistical language model to rank KPs, like the one presented in (Tomokiyo and Hurst, 2003).

However, the performance of the state of the art systems is still much lower than many other NLP tasks. An idea of the current top performing systems can be obtained by looking at the results of "SEMEVAL 2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles" (Kim et al., 2010), where systems were ranked by F-score on the top 15 extracted keyphrases. The best system, presented by (Lopez and Romary, 2010), achieved a score of 27.5%.

After SEMEVAL 2010, many systems tried to improve this level of performance, with an increasing focus over supervised systems. A common strategy is to look for new features to be used by ML algorithms. As an example, in (Haddoud et al., 2015) the authors were able to overcome the best SEMEVAL

This work is licenced under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/

performance achieving an F-Score of 28.6% on the top 15 KPs, by introducing a feature called Document Phrase Maximality, which they claim is able to better identify overlapping KPs, i.e. keyphrases that have a part in common, like for example "engineering" and "software engineering".

In this paper we follow the path of exploring new features and new ways of using linguistic knowledge from anaphora resolution to improve AKE. We started from the following hypotheses:

- If an n-gram is referenced many times inside a document, e.g. has many *anaphors*, its level of relevance as a KP may increase;
- If a pronoun can be replaced with the noun (or noun phrase) that it substitutes, we may detect information about said noun that otherwise would be lost; this information could be used to detect better KPs.

To check if these hypotheses hold we used the following approach. First, we set a baseline to compare our hypotheses against by choosing a minimal set of features that defined a system behaving like an average SEMEVAL 2010 contestant. Then, we designed two approaches, one based on the new linguistic features and the other based on a text preprocessing stage which applies anaphora-antecedent substitutions. Finally, we evaluated the performance of several ML algorithms using the SEMEVAL 2010 dataset and different feature sets combination which include the first hypothesis, the second one, or both.

2 Related work

The use of linguistic knowledge in AKE is not new. An interesting approach is the one presented in (Hulth, 2003), where the author wanted to demonstrate that the use of linguistic knowledge can lead to more compelling results than the ones obtained with the application of statistical features only. The AKE system proposed by Hulth exploits 4 features: three introduced in (Frank et al., 1999), which are within-document-frequency, collection-frequency and position of the first occurrence of the token, plus a new one that evaluates the POS-tag assigned to a term. The KPs extracted with the linguistic approach turned out to have lower recall but greater precision than the ones computed with the statistical one. In (Nguyen and Kan, 2007) the authors used a similar approach by taking into account also suffix sequences, acronym status and POS tag sequence as an hint for terminological status of the candidates. In (Pudota et al., 2010) instead the authors designed a system which weighted KPs based on the number of nouns contained in them. This system is the ancestor of the Distiller framework (Basaldella et al., 2015), which is the software we used in this work to perform our experiments.

Another way to improve AKE could be taking advantage of linguistic knowledge from anaphora and coreference resolution, which are the fields we are going to explore in this paper. Anaphora resolution is the problem of resolving what a pronoun or a noun phrase refers to. Lappin and Leass (1994) proposed "an algorithm for identifying both intrasentential and intersentential antecedents of pronouns in text": they use syntactical and morphological filters to find out dependencies between pronouns and possible related noun phrases (herein NPs), scoring the candidates by considering salience to select an adequate antecedent for each pronoun not flagged as pleonastic.[1]

A close field to anaphora resolution is *coreference resolution*. These two fields share similar information, so they overlap in a certain way: resolving anaphora is about finding the "cohesion"[2] which points back to some previous item" as stated in (Halliday and Hasan, 2014). So, the process of binding an antecedent to an anaphora is anaphora resolution; coreference resolution instead is the process of detecting when anaphors and their antecedent(s) have in common the same referent in the real world. Consider this the example from (Mitkov, 2014):

> This book is about anaphora resolution. The book is designed to help beginners in the field and its author hopes that it will be useful.

[1] A pleonastic pronoun is tipically used in phrasal verbs as subject or object but without an effective meaning (i.e. it seems, it is known, etc.)

[2] see section 3.1

Then the NP "the book" and both the pronouns "its" and "it" are anaphors referring to the antecedent "This book", and all three anaphors have the same real-word referent, which is "this book". So anaphors and their antecedent(s) are coreferential and form a chain, called *coreference chain*, in which all the anaphors are linked to their antecedent.

In our experiment we use the Stanford Coreference Resolution System `dcoref` (Manning et al., 2014) to retrieve anaphors and referents to implement our linguistics based features. We choose this software because of its good performance, since it is the direct descendant of the top ranked system at the CoNLL-2011 shared task. Moreover, both the Distiller and Stanford's system are Java-based, thus the integration of the two systems is easier. To resolve both anaphora and coreference, `dcoref` extracts the couples of anaphors and their relative referents, according to the matching of phrases' attributes, such as gender and number.

Other strategies to anaphora resolution, as the one introduced by Ge et al. (1998), use statistical information to resolve pronouns They use the distance between pronoun and the proposed antecedent to check the probability of the connection between them, information about gender, number, and animacity of the proposed antecedent as hints for the proposed referent, and head information to make selectional restrictions and mention count.

3 Anaphora in Keyphrase Extraction

3.1 Motivation

When we (humans) communicate, whether in a spoken or written form, generally we express a $coherent$ whole, i.e., a consistent and logical collection of words, phrases, and sentences. People often use abbreviated or alternative linguistic forms, such as pronouns, referring to or replacing some items that were previously mentioned in the discourse. Thus, to fully understand the discourse we need to interpret these elements, which *depend* on the elements they refer to. In linguistics, this process of interpretation is called *cohesion* (Mitkov, 2014).

On the other hand, when a document is processed for AKE, non influential words are usually removed. These words, commonly called *stop words*, are excluded from the text because they appear to be not significant, even if they are extremely frequent. Among them there are also pronouns such as *he*, *she*, *it*, *that*, *who*, and so on. The removal of such elements causes a loss of cohesion, both syntactically and semantically.

Moreover, pronouns have a relevant role in the sentences since they allow the author to enrich his writing using a richer vocabulary, composing more complex sentences, and so on. Pronouns are parts of the text which typically have the function of a substitute: depending on the case, they can replace a subject or an object, they can indicate possession, places, or refer back to people or things previously mentioned. Given these premises, disregarding all pronouns without replacing them with a valuable substitute could lead to a loss of a syntactical and/or semantical information. In fact, during the reading process we are able to decode the information conveyed by pronouns because we automatically replace them with the entity they refer to. In NLP a similar process is performed by anaphora resolution, thus our idea is to use this information, which would be otherwise lost, for AKE.

3.2 Definitions

We use the following definitions. Words are identified with the letter w and keyphrases with kp. Given a document d, $S(kp)$ is the set of sentences $s \in d$ in which kp appears. Given a sentence $s \in S(kp)$, we denote with $|s|$ the number of words in s, with $|kp|$ the number of words in kp, with $|S(kp)|$ the number of sentences in the set, and so on. Finally, $|C(d)|$ is the number of clauses in the document d, which are defined as a "simple sentences" or, more precisely, the smallest grammatical units which can express a complete proposition[3].

[3]Here we use *clause* and *proposition* as defined in (Kroeger, 2005).

3.3 First approach: Use of anaphora in Machine Learning features

As our first approach we decided to use linguistic knowledge to produce some new features. In detail, we designed a statistical feature that counts all the pronouns/pronominal anaphors which point to an entity (the *antecedent*), and a feature based on on lexical noun phrase anaphors, which are realized as definite noun phrases and proper names (Mitkov, 2014). We will call them *nominal anaphors* and *proper name anaphors* respectively.

For the first feature we follow this process: first, we use the Stanford CoreNLP Coreference Resolution System to find all the anaphors contained in a text and link them to their antecedent. Then, we select the pronominal anaphors, which are anaphors identified by personal pronouns *(he, she, ...)*, reflexive pronouns *(him, her, ...)*, possessive pronouns *(himself, itself, ...)*, demonstrative pronouns *(that, those, ...)*, and relative pronouns *(which, who, ...)*. Finally, we normalize the counted references for each antecedent dividing them by the number of clauses in the document.

Formally, we call $PA(kp)$ the set of pronominal anaphors for which kp is the antecedent. We define:

$$numOfReference(kp) = \frac{|PA(kp)|}{|C(d)|}$$

In our opinion, the use of sentences for normalization is not correct because within a sentence we could find more than one pronoun, skewing the normalization. If we choose the number of clauses to normalize the feature we are instead sure that $0 \leq numOfReference \leq 1$. For clarity, consider the "this book" example from (Mitkov, 2014) from in Section 2: by normalizing over sentences, the value of the feature would be $\frac{2}{1} = 2$, while by normalizing over clauses the value of the feature is $\frac{2}{5} = 0.4$.

The other linguistic feature we implemented is based on *nominal* and *proper name* anaphors. A nominal anaphora instead arises when the referring expression has a non-pronominal noun phrase as its antecedent: it is the case of clauses in which anaphora and antecedent are implicitly related, i.e., they do not stand in a structural or grammatical relationship, but they are linked by a strong semantic one. Consider this example from Wikipedia[4]:

> Margaret Heafield Hamilton (born August 17, 1936) is a computer scientist, systems engineer and business owner. She was Director of the Software Engineering Division of the MIT Instrumentation Laboratory, which developed on-board flight software for the Apollo space program. In 1986, she became the founder and CEO of Hamilton Technologies, Inc. in Cambridge, Massachusetts. The company was developed around the Universal Systems Language based on her paradigm of Development Before the Fact (DBTF) for systems and software design.

Here "Margaret H. Hamilton" is the $antecedent$ and the corresponding anaphors are the underlined words in the quote. "Computer scientist", "Director of the Software Engineering Division" are all examples of *nominal anaphors*.

The Wikipedia excerpt continues with this sentence:

> Hamilton has published over 130 papers, proceedings, and reports about the 60 projects and six major programs in which she has been involved.

Here, "Hamilton" is a proper name referring to "Margaret Heafiled Hamilton", and realizes a *proper name anaphora*.

When we read these sentences we automatically link for example the concept of being a "computer scientist" to a property of the subject of this sentence, while in AKE this information is lost. Hence, the basic idea behind this feature is to reward all the candidate KPs which appear in a nominal or proper name anaphora because they implicitly refer to the mentioned subject, highlighting important aspects of it.

[4] `https://en.wikipedia.org/wiki/Margaret_Hamilton_(scientist)`

In details, we process the document as previously defined. Then, for each candidate KP, we count all the times it appears in the set of the lexical noun phrases, i.e., the set of nominal and proper name anaphors. Finally we normalize the obtained score by the total number of appearances of the candidate in the document.

Formally, given a keyphrase $kp \in K$ and the set of the lexical noun phrase anaphors in the document NPA, the *inAnaphora* feature can be computed as follows:

$$inAnaphora(kp) = \frac{|\{a \in NPA | kp = a\}|}{|S(kp)|}$$

3.4 Second approach: Use of anaphora for preprocessing

While the previous approaches are able to capture some information about anaphora, they are not powerful enough to catch other knowledge that anaphora convey. For example, frequency-based features such as TF-IDF, which play an important role in several AKE algorithms since its first introduction in (Frank et al., 1999), may be recalculated using the anaphora in the frequency count as well.

This leads us to a different strategy: transforming the text into something that resembles the original human reading process as described in Section 3.1. To achieve this goal, we add to the our system a preprocessing stage that receives the original text from the dataset and substitutes in it all the non pleonastic pronouns with their antecedent. After this preprocessing phase, we perform AKE as usual and then we evaluate the results.

Consider the example we introduced in Section 3.3. If we apply the pre-processing, the sentence becomes:

> Margaret Heafield Hamilton (born August 17, 1936) is a computer scientist, systems engineer and business owner. Margaret Heafield Hamilton was Director of the Software Engineering Division of the MIT Instrumentation Laboratory, MIT Instrumentation Laboratory developed on-board flight software for the Apollo space program. In 1986, Margaret Heafield Hamilton became the founder and CEO of Hamilton Technologies, Inc. in Cambridge, Massachusetts. The company was developed around the Universal Systems Language based on Margaret Heafield Hamilton paradigm of Development Before the Fact (DBTF) for systems and software design.

Unfortunately, the original articles from the SEMEVAL 2010 Task 5 are transformed into plain text using the UNIX tool `pdftotext`. The output of this tool is a very unstructured text, where not only information about title, sections, etc., is lost, but also figures and tables may be placed inside content paragraphs, sentences may be badly split, and so on. This caused problems with the anaphora resolution and substitution algorithm, whose precision was undermined by these conversion errors. Moreover, these formatting problems may cause an erroneous coreference chain where the effects of an early bad resolution are amplified while going further in the text.

Thus, to improve anaphora resolution (and then substitution) in the original text, we segment the original text into sections. In this way, we improve the reliability of the parsing tree for the sentences and so we obtain more a correct performance in searching the antecedent. To perform this segmentation we use some heuristics to distinguish the title, the authors, and the email addresses, and to detect the boundaries of sections, paragraphs, figures, and so on. As a result, the text to process becomes more similar to the original graphical appearance in the PDF format, it is more structured, and it contains also less errors.

Then, by using this structure, we work on single sections, generating and using the coreference chain with the Stanford CoreNLP Coreference Resolution System to collect all the pronominal anaphors. Finally, we go back to the antecendent of each pronoun detected in the chain and we replace the former with the latter. The text turns out to be simpler but more informative for our AKE algorithm: by replacing the pronouns in the document we have no loss of information, while we are able to recover statistical and semantical information about the antecedents that would be otherwise lost.

The choice of substituting only pronominal anaphors is justified by the fact that nominal anaphors may not be just synonyms but also very different words, possibly with a different meaning. This happens because nominal anaphors have only the property of referring to the same entity in common thus substituting a nominal anaphora with its antecedent could change the meaning of the sentence. For example, from the text above, "Computer scientist", "system engineer", "Director of the Software Engineering Division", are all references to "Margaret Heafield Hamilton", but if we substitute them with the head of chain (i.e. "Margaret Heafield Hamilton"), the meaning of the sentence is completely different. Considering proper name anaphors is worthless as well: replacing a proper name anaphora with its antecedent could lead to a substitution that in our opinion could be useless or wrong. For example, in a biography there could be more people indicated by a common surname. Thus, an arbitrary substitution of all proper name anaphors could be wrong, because the anaphora resolution software may fail to identify the correct subject: in our example, if the head of the coreference chain is "Hamilton", we risk to replace "Margaret Heafield Hamilton" with her husband, whose surname is Hamilton too.

To summarize this approach, we follow this process: first, we parse the article from the dataset and use some heuristics to divide it into correctly formatted sections. Then, we process each section with the Stanford CoreNLP Coreference Resolution System, we collect all the pronominal anaphors, and we replace each anaphora with the correct antecendent. Finally, we submit the preprocessed text to the AKE process along with the value of the *InAnaphora* feature. Regarding to the *numOfReference* feature, which concerns only pronominal anaphors, its use has to be taken into consideration as well because when documents are preprocessed with substitution, different coreference chains could be discovered.

4 Methodology

4.1 Baseline algorithm

In order to evaluate the impact of the proposed features, we used the Distiller framework to implement a baseline keyphrase extraction algorithm with few basic features. In our baseline algorithm candidate KPs are n-grams selected from the text if they match a given set part-of-speech patterns, which is one of the most common way of generating candidates in literature (Hasan and Ng, 2014).

The baseline feature set for our experiment is a set of well-accepted features for AKE, i.e., given a candidate KP, we consider:

- TF-IDF;
- relative position of the first appearance of the candidate (*height*);
- difference between the position of the last and the first appearance (*lifespan*);
- number of appearances of the candidate in the text, normalized by number of sentences.

Then, we consider a new feature set, in which we add to the baseline a fifth feature called Document Phrase Maximality (DPM), introduced by Haddoud et al. (2015). We use this feature because it supposedly should help to discriminate between candidate keyphrases which often appear as substring of another candidate. We deem this feature as necessary because, by using our substitution algorithm, we usually substitute an anaphora with a longer antecedent, thus leading to an increase of frequency of all the words contained by the antecedent. In our example, we will substitute the anaphors with "Margaret Heafield Hamilton", thus increasing the frequency, e.g., of the word "Hamilton", but DPM allows us to assign a *low* score to the single words while assigning an *high* score to the full name of the scientist.

The Machine Learning algorithms we choose are logistic regression, neural networks, and boosted decision trees, since these algorithms have a reputation of being good algorithms in the AKE community (Hasan and Ng, 2014). We used their implementation with the R software, using the `glm`, `nnet` and `C5.0` libraries to train the respective models. We used no particular tweaking on the algorithms; the neural network used was a simple Multi Layer Perceptron (MLP) with one hidden layer. Then, we ranked KPs using the raw probability output by the algorithms.

																P	R	F1	MAP
A	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	0.33	0.33	0.33	0.33
B	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	0.33	0.33	0.33	0.02
C	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	0.40	0.40	0.40	0.31

Table 1: Hypothetical Precision, Recall, F1-Score and MAP of three systems A, B and C over a document with 15 correct keyphrases. A tick mark (✓) indicates a system assigned correct keyphrase, while an x mark (✗) a wrong one.

We didn't choose a bigger feature set because there is no agreement on which are the "state of the art" features for AKE. The top performing systems use many different strategies: in (Lopez and Romary, 2010), the authors use few features, but a custom "post-machine learning" ranking algorithm; in (You et al., 2013) few features are used, but a different candidate generation algorithm; in (Haddoud et al., 2015), we can find more than 20 features.

Moreover, this simple feature set is enough to get an average performance on the SEMEVAL task. With the MLP, our baseline system showed an F-score of 19.69% on the best 15 keyphrases, which is good enough to be ranked 11^{th} out of 20 contestant in the SEMEVAL 2010 challenge. The same position would be achieved using logistic regression, with a score of 19.22%, while the use of decision trees causes a slip of one position down, with a score of 18.95%.

4.2 Metrics

The usual metrics used in AKE are Precision (P), Recall (R), and F1-Score, but these metrics are not the only ones that we will consider in evaluating our system. In fact, we believe that our proposed system may offer an interesting contribute even if we are just able to provide a *better ranking* of the keyphrases.

As pointed out in (Schluter, 2015), this better ranking could not be caught by the aforementioned metrics. For example, suppose we have two systems participating in SEMEVAL 2010, where algorithms are ranked by the F1-Score of the top 15 keyphrases returned by the algorithms. We call this systems A and B. Suppose that, looking at A's output, only the first 5 keyphrases are correct, while the other 10 are wrong. Then, suppose that B's output is the opposite: the first 10 keyphrases are wrong, then the next 5 are good. So, this system will have the same precision (33%), the same recall and the same F1-Score, but the ranking of the system A is arguably better than the ranking of the system B.

Therefore, to evaluate our system, we propose the use of the Mean Average Precision (MAP) metric, which is more suitable to evaluate a ranking than simple precision and recall. We borrow our definition of MAP for keyphrase extraction from (Manning et al., 2008), that is, if the set of correct keyphrases for a document D is $\{kp_{(D,1)}, \ldots, kp_{(D,n)}\}$ and $R_{D,k}$ is the set of retrieved keyphrases for the document D until you get to the k-th keyphrase,

$$MAP(D,R) = \frac{1}{n}\sum_{j=1}^{n}\frac{1}{j}\sum_{k=1}^{j} Precision(R_{D,k})$$

Where $Precision(R_{D,k})$ is the Precision@k score of the system over document D for the first k retrieved documents.

As an example, we see the systems A and B compared in Table 1, on a document with 15 gold keyphrases. The two systems share the same P, R and F1 scores, while MAP is radically different, being much higher for system A than for system B. If we suppose to have another system C, which gets the 1^{st},2^{nd},3^{rd},5^{th},10^{th} and 15^{th} keyphrases correct, we have that this system shows higher P, R and F1 scores than systems A and B but lower MAP than A. This happens because of A's better quality of the first results or, in other words, because of the higher *"weight"* of A's correct keyphrase in the fourth position than C's correct keyphrases in tenth and fifteenth position.

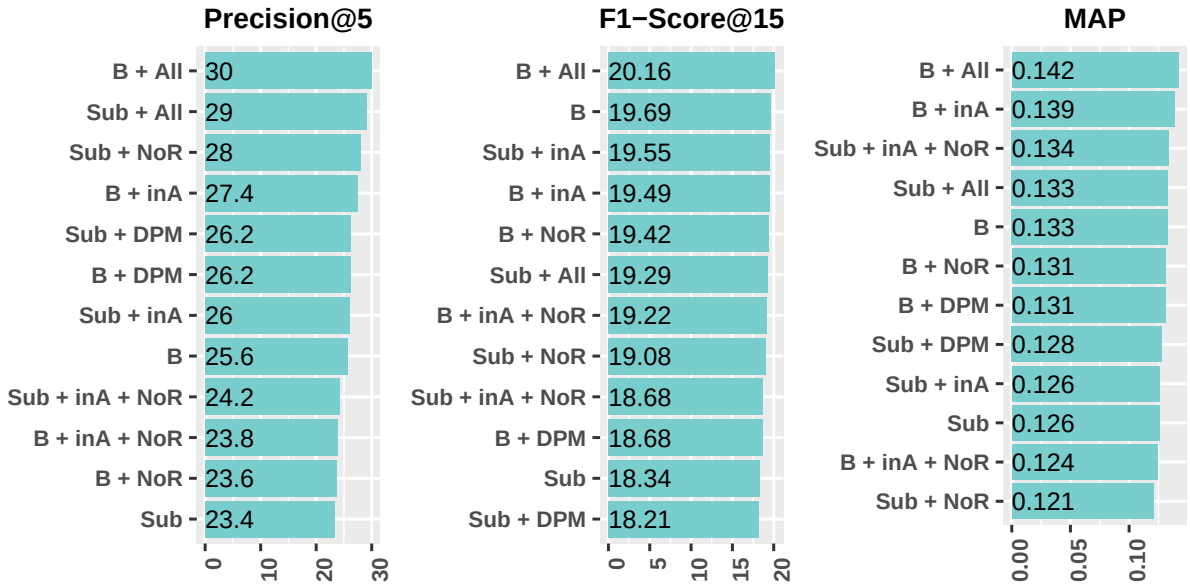

Figure 1: Scores obtained by running our keyphrase extraction algorithm over different feature sets. Note that "B" stands for "Baseline", "Sub" indicates the baseline features ran on the preprocessed documents, "inA" marks the $inAnaphora$ feature, "NoR" marks the $numberOfReference$ feature, "DPM" stands for the Document Phrase Maximality feature, and "All" indicates that the feature set contains all the aforementioned features.

5 Results

Combining the baseline features defined in Section 4.1 with our new features defined in Section 3.3 and the pre-processing technique we described in Section 3.4, we defined a total of 36 different AKE pipelines. These pipelines were built by running the three machine learning algorithms we choose on baseline feature set first, then adding DPM and our anaphora-based features, then combining the features together, both on the original SEMEVAL 2010 dataset and the text preprocessed with our technique.

The neural network was the best performing algorithm overall, and we can see its results in Figure 1. The first impression is that there is generally a little improvement in F1-Score, with the baseline algorithm on the original documents still being the second best feature set with this metric.

Nevertheless, looking at the Precision@5 score, we see that our approach has a significant impact: as shown in Figure 1 (left column), the combination of the linguistic features with the statistical ones, both in the original documents and in the ones with preprocessing, the precision score is greater than the one obtained for the baseline set. In particular, starting from the result of 25.60% for the baseline, we reach a score of 27.60% in precision just by using *inAnaphora* feature and 30.00% adding also *numOfReference* and *DPM*. A similar behavior can be observed in the results when considering that the substitution technique is performed in the preprocessing. In fact, while on the preprocessed text the baseline features show no improvement, a more interesting result can be seen using the linguistic features, for which precision score raises from the baseline's 23.40%, to 26.00% adding *inAnaphora*, and to 29.00% combining all the features together.

Looking at the scores of MAP, we can see that using all features on the original dataset offers the best ranking, as it would be expected by the high Precision@5 score; this confirms our idea of combining the anaphora-based features with DPM. Interestingly enough, most of the other feature sets show a slight decrease in the quality of the ranking, probably because the gain in precision is not high enough to balance the decrease in recall.

Taking into account just the second approach, the results provide evidence of our initial assumptions on the importance of using *inAnaphora* feature over preprocessed text. Precision and F1-Score show a more significant increment when using preprocessed documents, and the reason can be found in a second

parsing with more *coherent* coreference chains. In details, coreference resolution and so our features that depend from it improve because the substitution of pronouns with the common antecedent in the first chain produces a text with more noun phrases and less pronouns. This way, the parse tree of the preprocessed text is simpler, so the relationships between the "new" noun phrases are more clear. This allows the anaphora resolution library to find more anaphors and to better detect pleonastic pronouns, thus obtaining a more precise score for our feature.

The other ML algorithms (not shown in the figures) seem to prove the conclusions we obtain from the neural network: using either decision trees or logistic regression the behavior is similar to the one described for the neural network, with a relatively stable F1-Score on the top 15 extracted keyphrases, but a significant increase in Precision@5 and MAP score when adding linguistic features. It is interesting that for both algorithms, using all features on the original documents offers highest Precision@5 and MAP scores, confirming the results shown in Figure 1. In particular, with this features/dataset combination, with the `glm` library we see slight rises in Precision@5, F1-Score@15 and MAP from 24% to 24.4%, from 19.22% to 20.30% and from 0.127 o 0.136 respectively; for `C5.0`, while F1-Score rises from 18.95% to just 19.29%, Precision@5 and MAP shows a more significant improvement from 22.2% to 26.8% and from 0.123 to 0.137 respectively, thus supposedly showing a better ranking of the keyphrases found. On the other hand, using the same feature set over the preprocessed documents still shows an improvement from the baseline, but with slightly lower scores.

6 Conclusions

Our analysis shows that anaphora and coreference resolution can be used for AKE with significant results. Like in (Hulth, 2003), we see that by exploiting linguistic knowledge in a keyphrase extraction algorithm it is possible to increase the precision of the results. We think that it is important to analyze the relationships which could arise when linguistic features are combined together with statistical features. For example, it is clear that preprocessing the input text by substituting the pronouns with the entity they refer to could increase the frequency of certain terms, thus statistical features like DPM can be useful to gain a performance boost.

A better result could be obtained by improving anaphora resolution performance, since the software we used was not always able to find all the correct anaphors, even if it is (or it is close to) the state of the art system for anaphora and coreference resolution at the time of writing. For example, looking at the example we introduced in Section 3.3, the algorithm was not able to detect the anaphora *director* from the sentence "*She was Director of the Software Engineering Division*", which means that we would not able to detect and replace correctly all the pronouns in the coreference chains or compute the value of our features correctly. This is confirmed by the fact that the *numOfReference* feature, which is based on the count of pronominal anaphors, had a positive impact on performance even after the text preprocessing, which *should* have had replaced all the pronouns with their antecedents.

As a future work, we consider the idea of using the outcomes of our preprocessing stage for improving anaphora resolution specifically for the task of keyphrase extraction, developing an ad-hoc mining algorithm for the parsing trees, with the goal of producing a better pre-processing algorithm and finding for each valuable pronoun a good candidate antecedent. Moreover, another interesting approach would be looking for statistical features other than DPM which are able to better interact with the anaphora-related ones.

References

Marco Basaldella, Dario De Nart, and Carlo Tasso. 2015. Introducing Distiller: a unifying framework for knowledge extraction. In *Proceedings of 1st AI*IA Workshop on Intelligent Techniques At Libraries and Archives co-located with the XIV Conference of the Italian Association for Artificial Intelligence (AI*IA 2015)*. Associazione Italiana per l'Intelligenza Artificiale.

Dario De Nart, Dante Degl'Innocenti, Andrea Pavan, Marco Basaldella, and Carlo Tasso. 2015. Modelling the user modelling community (and other communities as well). In *User Modeling, Adaptation and Personalization:*

23rd International Conference, UMAP 2015, Dublin, Ireland, June 29 – July 3, 2015. Proceedings, pages 357–363. Springer International Publishing.

Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Domain-specific keyphrase extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, IJCAI '99, pages 668–673. Morgan Kaufmann Publishers Inc.

Niyu Ge, John Hale, and Eugene Charniak. 1998. A statistical approach to anaphora resolution. In *In Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–170.

Carl Gutwin, Gordon Paynter, Ian Witten, Craig Nevill-Manning, and Eibe Frank. 1999. Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems*, 27(1):81–104.

Mounia Haddoud, Aïcha Mokhtari, Thierry Lecroq, and Saïd Abdeddaïm. 2015. Accurate keyphrase extraction from scientific papers by mining linguistic information. In *Proceedings of the Workshop Mining Scientific Papers: Computational Linguistics and Bibliometrics, 15th International Society of Scientometrics and Informetrics Conference (ISSI), Istanbul, Turkey*.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. 2014. *Cohesion in english*. Routledge.

Kazi Saidul Hasan and Vincent Ng. 2014. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1262–1273. Association for Computational Linguistics.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223. Association for Computational Linguistics.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26. Association for Computational Linguistics.

Paul R Kroeger. 2005. *Analyzing grammar: An introduction*. Cambridge University Press.

Shalom Lappin and Herbert J Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational linguistics*, 20(4):535–561.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 257–266. Association for Computational Linguistics.

Patrice Lopez and Laurent Romary. 2010. HUMB: Automatic key term extraction from scientific articles in GROBID. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 248–251. Association for Computational Linguistics.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *An Introduction to Information Retrieval*. Cambridge University Press.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Ruslan Mitkov. 2014. *Anaphora resolution*. Routledge.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326. Springer.

Nirmala Pudota, Antonina Dattolo, Andrea Baruzzo, and Carlo Tasso. 2010. A new domain independent keyphrase extraction system. In *Digital Libraries: 6th Italian Research Conference, IRCDL 2010, Padua, Italy, January 28-29, 2010. Revised Selected Papers*, pages 67–78. Springer Berlin Heidelberg.

Natalie Schluter. 2015. A critical survey on measuring success in rank-based keyword assignment to documents. *22eme Traitement Automatique des Langues Naturelles, Caen*.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*, MWE '03, pages 33–40. Association for Computational Linguistics.

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

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 855–860.

Wei You, Dominique Fontaine, and Jean-Paul Barthès. 2013. An automatic keyphrase extraction system for scientific documents. *Knowledge and Information Systems*, 34(3):691–724.