Keyword Extraction for Text Characterization

Ingrid Renz, Andrea Ficzay, Holger Hitzler

Information Mining
DaimlerChrysler AG, Research and Technology
P.O. Box 23 60
89013 Ulm, Germany
ingrid.renz@daimlerchrysler.com

Abstract: Keywords are valuable means for characterizing texts. In order to extract keywords we propose an efficient and robust, language- and domain-independent approach which is based on small word parts (quadgrams). The basic algorithm can be improved by re-examining and re-ranking keywords using edit distance (i.e. Levenshtein distance) and an algorithm based on the relativistic addition of velocities (here: weights). For the purpose of evaluation, we compare our approach to frequency-based keyword extraction (exemplary text collection: 45000 intranet documents in German and English).

1. Keywords for Text Characterization

The analysis of huge text collections usually aims at finding relevant texts (known as text retrieval with search engines) or text groups (supervised grouping like categorization or classifying, unsupervised grouping like clustering). Thus, all these text mining tasks result in retrieved texts or text groups. For an overview of text mining tasks see [Ch00].

But it is a tedious task of any information-seeking user to scan all retrieved items. In order to facilitate this task, most text mining systems characterize their resulting texts with various kinds of annotations. They link texts to external topic schemes or find relevant concepts out of the texts themselves. These items of external topic schemes as well as text-based concepts can be presented as keywords which are a helpful characterization of textual content.

2. Previous Work

An extensive survey of summarization gives [Ho02]. Here, topic identification as the simplest type of a summary also subsumes keyword extraction.

As best-known approaches, he lists:

- position method which defines relevant words according to their text position (heading, title),
- cue phrase indicator criteria (specific text items signal that the following/previous words are relevant),
- frequency criteria (words which are infrequent in the whole collection but - relatively - frequent in the given text, are relevant for this text),
- connectedness criteria (like repetition, co-reference, synonymy, semantic association).

These approaches differ according to the resources they need and to the universality they can be applied to. Position method surely is restricted to specific text genres. Cue phrase indicator criteria as well as connectedness criteria need language-specific resources: in the first case, simple lists may be sufficient, but in the second case, an in-depth linguistic analysis is needed.

Only topic identification based on frequency criteria can be generally applied in any given text collection independently of the language used.

3. Quadgram-Based Keyword Extraction

In [BF99] and [BD00), we described our work concerning an intranet. Further text collections we had to analyze include customer feedback statements, accident reports, or problem descriptions of user help desks (see [BD02]). These text collections differ not only in the languages but also in the structures the single texts are written in.

Nevertheless, it is a common task in all these collections to characterize a text with representative words. Thus, our goal is an efficient and effective keyword extraction method which is language-, domain-, genre-, and application-independent.

3.1 General Setting: Vector Space Model

In information retrieval ([SM83]), the vector space model is widely used for representing textual documents and queries. Given a text collection, a set of terms or features has to be defined. Then, for each text a (high-dimensional, sparsely filled) vector is generated from this feature set with associated weights. The weights are usually computed by measures like tf/idf (i.e. terms frequency in given text times inverse document frequency in whole collection).

3.2 Quadgrams as Features

The most common text features are (a subset of all) words the considered texts consist of. When these words are fractionalized into overlapping strings of a given length N (here: 4), N-grams (here: quadgrams) result. For example, the word *feature* consists of 6 quadgrams: *_fea, feat, eatu, atur, ture, ure_*. As consequence, the completely different word strings *features* and *feature* have 5 quadgrams in common (and are therefore similar in the vector space model). N-grams are tolerant of textual errors (see [CT94]), but also well-suited for inflectional rich languages like German. Computation of N-grams is fast, robust, and completely independent of language or domain. In the following, we only consider quadgrams because in various experiments on our text collections, they have overtopped trigrams.

3.3 Quadgram-Based Keywords

In order to represent a text in the vector space model, its words and - in the following more important - its quadgrams were associated with their tf/idf-weights (given that all words and quadgrams are used as features). Now, the basic idea for extracting keywords is that weighted quadgrams indicate relevant words. For example, given that in a text quadgrams like *cate, tego, izat* are high-weighted, then words like *categorization* (3 hits), *categorize*, *category* and *categories* (2 hits), which contain these quadgrams, are relevant.

In order to compute the weight of a keyword, we can add up the tf/idf-weights of all (*n*) quadgrams the word contains. But this favors longer words which consists of more quadgrams. Therefore, a normalization is needed.

Normalization to word length:

$$Weight_{Word} = \frac{\sum_{i=1}^{n} Weight_{Quadgram}(i)}{Length_{Word}}$$

strongly penalizes long words. Hence,

Normalization to logarithm of word length:

$$Weight_{Word} = \frac{\sum_{i=1}^{n} Weight_{Quadgram}(i)}{\log(Length_{Word})}$$

adjusts this drawback. Since usually the number of quadgrams in a text collection is too big to be handled in the vector space model, feature selection chooses only the most relevant quadgrams (by their tf/idf-weights) as features. Therefore, not all quadgrams a word consists of is part of the feature set and can be used for keyword extraction. This fact is exploited for normalization. As measure for the selection of keywords we propose the sum of the weights of quadgram features contained in a word divided by the number of quadgrams features (n) in this word.

Normalization to number of quadgram features:

$$Weight_{Word} = \frac{\sum_{i=1}^{n} Weight_{Quadgram}(i)}{n}$$

Experiments have shown that this normalization provides better keywords given an appropriate quadgram feature selection (for a discussion of quality aspects see section 4). If such a feature selection is not possible, normalization of logarithm to word length should be applied.

3.4 Re-Examination of Keywords

Independently of the normalization algorithm chosen, the presented quadgram-based keyword extraction suffers from "word similarities". The above example shows that high-weighted quadgrams often appear in (inflectionally or morphologically) related words. In order to optimize the information content of the keywords presented, these related words must be identified and only one representative of them should be presented to a user.

A simple, fast, and general algorithm to identify similar word forms is the edit- or Levenshtein-distance ([Le75]). Here, the number of edit-operations (deleting or inserting one character) which are needed to transform one word form (W_1) into another (W_2) is counted. When computing the word distance (D_{Lev}), the lengths of the word forms have to be considered, too.

$$D_{Lev}(W_1, W_2) = \frac{\sum EditOps(W_1, W_2)}{Length(W_1) + Length(W_2)}$$

If two keywords (W_1, W_2) have a low distance (here, a - parameterizable - threshold must be set), the lower rated (or shorter, longer, less frequent) keyword is removed.

But how to handle the weight of the remaining keyword? Certainly, its weight should be augmented in order to incorporate the weight of the removed keyword. Computing the mean weight reduces the original weight of the higher-weighted keyword. Adding both weights may overshoot the scope of weights ([0,1]). Here, we use a formula which is based on Einstein's velocity addition relationship. It has the needed properties to augment the weight without exceeding the scope:

$$Weight_{total} = \frac{Weight_1 + Weight_2}{1 + Weight_1 * Weight_2}$$

4 Results

In order to evaluate our quadgram-based keyword extraction we compared it to the other resource-free, language- and domain-independent extraction method: the one based on frequency criteria, i.e. for a considered text, we took its words with highest tf/idf-measure as keywords.

Algorithmic properties: Since the computation of tf/idf-values for words is a basic part of our systems, words of a given text only have to be sorted according to these values ($O(w_i*log(w_i))$, w_i = number of words in text i). Then, the top N words are chosen as tf/idf-keywords. Also the tf/idf-values for quadgrams are available, but here, any word has to be fragmented into its quadgrams ($O(w_i)$), weighted, re-examined and sorted ($O(w_i*log(w_i))$). As exemplary time effort (on Pentium II, 400 MHz): for the analysis of 45000 intranet documents, the step of keyword extraction needs 115 seconds using tf/idf-keywords and 848 seconds using quadgram-based keywords.

Quality of keywords: A general measure similar to recall/precision in information retrieval can hardly be defined since the users' judgements about quality are very subjective. Inspecting keywords of concrete texts show that among the top 10 keywords approximately 30% are tf/idf- as well as quadgram-based keywords. 70% are different: quadgram-based keywords seem to be more text-specific, tf/idf ones prefer proper names. For a given text[1] , the following top 10 keywords have been computed:

[1] A-Klasse laut ADAC: "Mercedes A-Klasse: Keine Probleme mit dem Elch" ADAC untersucht Kippverhalten von Fahrzeugen. Stuttgart-Möhringen, 7. November 1997. Die A-Klasse von Mercedes ist, was ihr Kippverhalten angeht, besser als ihr Ruf. Ein jetzt vom ADAC durchgeführter Fahrversuch hat gezeigt, daß der kleine Benz bei Slalom- und Spurwechseltests stärker als vergleichbare Fahrzeuge wankt, unter normalen Testbedingungen aber beherrschbar bleibt. Das trifft auch für den sogenannten Elchtest zu, mit dem die A-Klasse in die Schlagzeilen gefahren ist. Bei den ADAC Fahrversuchen kam das Fahrzeug im Gegensatz zu den mitgetesteten Konkurrenten tatsächlich an die Kippgrenze. Dies trat allerdings nur mit den Reifen auf, die von Mercedes mittlerweile zurückgezogen wurden. Mit den inzwischen vom Werk freigegebenen Michelin-Reifen schaffte der Mercedes den Slalom- und Elchtest sowohl leer als auch vollbeladen und zeigte sich den gleichzeitig getesteten Konkurrenten VW Golf, Renault Megane Scenic und Citroen Xsara gleichwertig. In ihrer Dezember-Ausgabe wird die ADAC-Motorwelt ausführlich über einen Vergleichstest mit den genannten Fahrzeugen berichten.

quadgram-based	tf/idf- keywords
adac	a-klasse
fahrversuch	mercedes
fahrzeug	konkurrenten
mercedes	vergleichbare
a-klasse	leer
elchtest	gefahren
elch	vw
untersucht	ausführlich
vergleichstest	möhringen
konkurrenten	fahrzeuge

Context sensitiveness: We investigated which keywords will be computed if the text is situated into different collections. As experimental set-up we took from the original exemplary collection (45000 texts) arbitrarily 10000 texts (set 1) and 2000 texts (set 2). Analyzing keywords of texts which belong to each set showed that quadgram-based keywords are quite insensitive to their collection, i.e. barely any difference is found. In contrast, tf/idf-keywords change to a greater extent. This is a consequence of the used tf/idf-measure where document frequency of a word directly plays a crucial role. For quadgram-based keywords the document frequency of quadgrams is - only indirectly - used. In the following table, top 10 keywords of the given text (see footnote 1) as part of set 2 (2000 texts) are listed:

quadgram-based	tf/idf- keywords
adac	a-klasse
fahrversuch	mercedes
fahrzeug	trifft
a-klasse	reifen
mercedes	fahrzeugen
elchtest	vw
elch	beherrschbar
slalom	angeht
untersucht	zeigte
vergleichstest	ausführlich

Keywords as search terms: In order to prove the coverage of keywords, they were used as queries in search engines. If an internet search is performed, tf/idf-keywords will get better hits. Here, quadgram-based keywords are too text-specific. But searching the intranet, where the text belongs to, returns better hits for the quadgram-based keywords than for the tf/idf-ones which are not accurate enough.

5 Conclusion

It can be stated that quadgram-based keywords are as easily computed as frequency-based. Both approaches are solely syntactic and do not need expensive resources or analysis. Therefore, they are completely language-, genre- and domain-independent.

We applied keyword extraction to texts of various collections (customer feedback statements, intranet documents, news articles) written in German and English. These text collections differ not only in language, but also in length (from 8 to 5000 words) and in quality (near to spoken language, highly elaborated written texts).

In comparison to tf/idf-keywords, quadgram-based keywords are more text-specific and more text-oriented. In our different applications, they proved to be valuable for characterizing texts.

References

[BF99] Bohnacker, U.; Franke, J.; Mogg-Schneider, H.; Renz, I.; Veltmann, G.: Restructuring Intranets by Computing Text Similarity. In (Sandrini P. ed.): TKE99 - terminology and Knowledge Engineering. Termnet, Wien, 1999; 610-617.

[BD00] Bohnacker, U.; Dehning, L.; Franke, J.; Renz, I.; Schneider, R.: Weaving Intranet Relations - Managing Web Content. In Proc. of RIAO2000: Content-Based Multimedia Information Access, Paris 2000; 1744-1751.

[BD02] Bohnacker, U.; Dehning, L.; Franke, J.; Renz, I.: Textual Analysis of Customer Statements for Quality Control and Help Desk Support. In (Jajuga, K.; Sokolowski, A.; Bock, H. eds.): Classification, Clustering, and Data Analysis. Recent Advances and Applications. Springer, Berlin, 2002; 437-445.

[CT94] Cavnar, W.; Trenkle, J.: N-Gram-Based Text Categorization. In Proc. of Symposium on Document Retrieval and Information Retrieval, Las Vegas 1994; 161-175.

[Ch00] Chakrabarti, S.: Data Mining for Hypertext: A Tutorial Survey. In ACM SIGKDD Explorations, 1:2, 2000; 1-11.

[Ho02] Hovy, E.: Automated Text Summarization. In (Mitkov, R. ed.): Oxford University Handbook of Computaltional Linguistics. Oxford University Press, Oxford, 2002.

[Le75] Levenshtein, V.: On the Minimal Redundancy of Binary Error-Correcting Codes. In Information and Control, 28:4, 1975; 268-291.

[SM83] Salton, G.; McGill, M..: Introduction to Modern Information Retrieval, McGraw Hill, 1983.