Recording word position information for improved document categorization

Piotr Gawrysiak[1], Lukasz Gancarz, Michal Okoniewski

Institute of Computer Science, Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
{gawrysia, gancarz, okoniews}@ii.pw.edu.pl

Abstract. In this paper, which is a report from work in progress, we briefly present the new document representation that could be used in classic text mining applications, such as document categorization. We briefly present most popular unigram and n-gram document representations that are used frequently in text mining research, mention their shortcomings, and present the idea of representation that records not only word count information, but also word position within document. As experiments are still underway, we do not present final result, but only mention types of tests that are being done.

1 Preliminaries

Automatic document categorization (or classification) has become quite recently one of the very popular areas of research. As with other text mining methods, this is rather case of re-discovery, related mostly to the exploding popularity of the World Wide Web, and very poor quality of existing information retrieval and document management tools. The advent of so called *knowledge management* methodologies is probably also an important factor here, as large organizations nowadays tend to use automated information and document management systems. Automatic document categorization belongs to the statistical text processing methods, which in general deliver much better results (in terms of practical quality of output in business environments) than classic language analysis methods. Automatic document categorization is also a data mining method, and therefore owes much to the data mining hype of the recent years. Detailed discussion concerning popularity and usefulness of DM related text analysis tools is of course beyond the scope of this paper. For further information consult, for example [1], or the thesis [2].

Document categorization (or categorization in general) is, surprisingly, often confused with clustering (as in [3]). Therefore a short definition may be appropriate. Let $D=\{d_1, d_2, ..., d_n\}$ be a set of text documents and $K=\{k1, k2, ..., k_l\}$; $K|\geq 2$, $|K|\leq|D|$ a set of identifiers, defining *classes*. Assignment $f: D\rightarrow K$ defines the document – class relationship. Let T be a subset of $D, T \subset D$, $T=\{t_1, t_2, ..., t_m\}$ called a *training set*. Document categorization is a process of estimating the assignment g, based on values

[1] Part of this work has been financed by EU Marie Curie Programme HPMT-CT-2000-00049

of an assignment *f* for elements from *T*, while the differences between *g* and *f* should be minimized for entire set *D*. Finally if $|K| = 2$ the categorization is said to be *binary* and if $\exists k_a, k_b : \forall d_i \ f(d_i) = k_a \Rightarrow f(d_i) = k_b$ the categorization is hierarchical.

In other words in a categorization process a computer system tries to assign previously unknown documents to classes, basing on the database of documents (the *training set*) that have been already classified, for example by human analyst.

As we mentioned above, the document categorization owes a lot to classic data mining research. In fact most of the existing categorization systems directly use classification algorithms, designed for relational databases (and therefore they are able to process only data consisting of pairs attribute - value). Such algorithms cannot be of course directly applied to text documents, so some kind of conversion is necessary, where characteristics of documents are captured in a specific *document representation*, as Figure 1 illustrates.

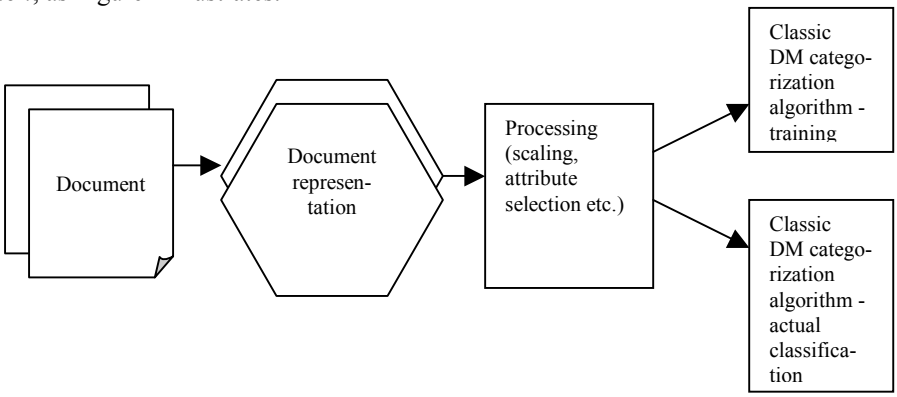

Fig. 1. Structure of a typical document categorization system

Practically all currently popular classification algorithms have been used in document categorization systems. These include Naive Bayes [4], [5], decision trees [6], neural nets, kNN and finally SVM [7], [8] which seem to achieve best performance on standard test corpora. The paper [9] compares the performance of above algorithms applied to text data, concluding that SVM , kNN and Rocchio obtain best results, while worst performance seems to be achieved by Naive Bayes, and Neural Nets.

No matter how sophisticated and robust the classification algorithm is, if the document representation is poor, the results produced by entire system will be also flawed. For example if one decided to use a very crude representation consisting only of one attribute, the entire system would be completely useless, even if the actual classification were performed by SVM algorithm. It is therefore a bit surprising that relatively little research concerning document representations takes place nowadays. Most work in improving document categorization systems seems to be focused on representation processing, and of course on classification algorithms. There are important exceptions of course, such as [10] where the usefulness of simple n-gram representations is discussed or [11] proposing interesting thesaurus-based representation, to name a few. However in most cases a simple *bag of words* or *unigram* representation is used. This

representation is simply based on counting word occurrences producing for each document $D=(w_1, w_2, ..., z_1, ..., w_n, z_m)$ a vector $\boldsymbol{R}$ such that

$$Rbin_i = \begin{cases} 1 \; if \; \exists j; w_j = v_i, v_i \in V \\ 0 \; otherwise. \end{cases} \quad \text{in } \textit{binary unigram}^2 \text{ or} \tag{1}$$

$$Rmlt_i = \frac{\sum_{j=1}^{n} \begin{cases} 1 \; if \; w_j = v_i, v_i \in V \\ 0 \; otherwise \end{cases}}{n} \quad \text{in } \textit{multivariate unigram}. \tag{2}$$

One must ask the question, whether more complex representations than *unigram* would possibly yield any useful results. Apparently, the performance of modern categorization systems seems to be very good. Their quality, measured as precision-recall breakeven point over standard Reuters-21578 test corpus [12], comes close to 90 percent (see for example [13]), what is indeed impressive. However in our opinion many of the currently used test corpora - such as Reuters or Digitrad, have serious drawbacks. First, only pure text documents are contained within, while in contemporary business world the documents being processed are quite richly formatted. This may be true even in case of simple office memos. Next, the class hierarchies in these repositories are very large, consisting often of several hundred classes, while at the same time the class boundaries are very well defined - for example for several classes in Reuters corpus testing for the presence of even one or two keywords can be sufficient for correct class assignment for a new document. In practical industrial applications the situation is often reversed - the number of classes is quite small, but document assignment criteria are much more complex. Finally, and perhaps most importantly, the documents from these corpora are very short. For example a lot of Reuters documents contain only a handful of sentences. Such documents are very rarely encountered in business environment, where most document management tasks concern large reports, specifications and press articles, which are in general much longer. Therefore while experimental systems are reported to achieve high categorization performance, the popularity of such tools in commercial applications still remains limited.

It is however possible, that more complex document representations, where not only word count is stored, but also information about word position, or even it's formatting, could be useful in analysis of such longer documents. As a crude example consider the text of Lewis Carroll's "Through the looking glass"[3] (which is of course not a business document, but it is long and not limited in scope). Let's analyse the occurrences of words "any" and "dumpty". Both these terms have the same occurrence frequency - there are 53 occurrences in the entire text. Therefore for the categorization system based on *unigram* representation these two words are equally significant.

[2] This terminology comes from paper [McCallum, 1998].

[3] The etext version [Caroll, 1994] has been used.

However if we analyse the positions of these occurrences in the text, we can conclude that these two keywords in fact belong to different semantic categories.

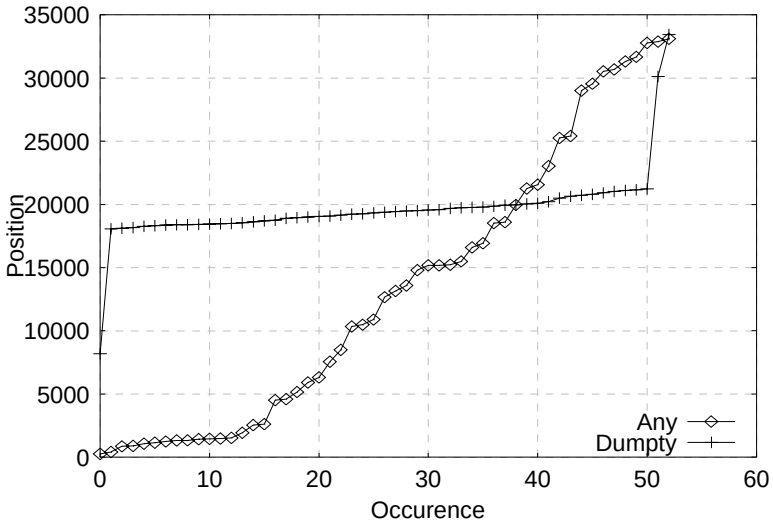

Fig. 2. Occurrences of words any and dumpty in [14]

Another example involves analysis of positions of the same (or similar) terms in two different documents. There is a strong indication that these two documents belong to the same class if the occurrence patterns are similar (for example if the terms occur only in the beginning of the documents and near its ends). Figure 3 illustrates such simple comparison for two documents extracted from Project Gutenberg repository. Documents not belonging to this repository, but only describing it (and hence also containing the terms “etext” and “project”) would have in most cases different occurrence pattern. In fact such simple analysis may be even useful in information retrieval system, as it has demonstrated by a bit forgotten *TileBars* search results visualization system [15].

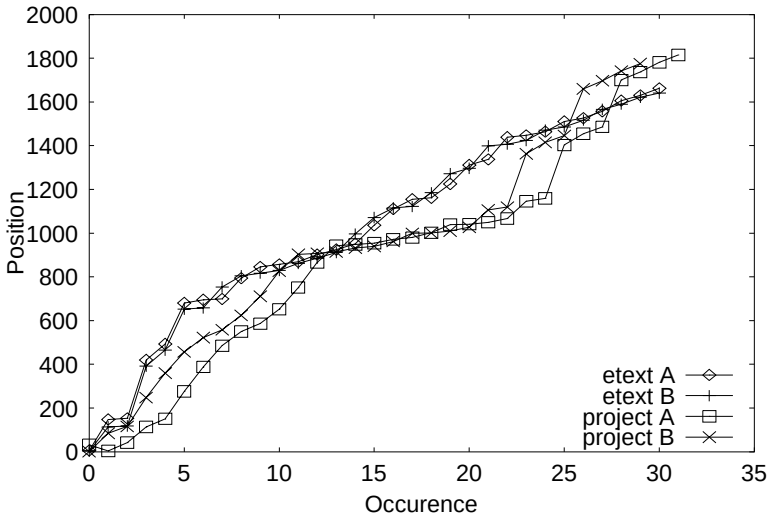

Fig. 3. Occurrences of words project and etext in two documents

Such generalisations may be of course dangerous, and in most cases a simple unigram-based categorization system would be able to properly categorize the documents from above examples. However, it would do so using different clues that would be utilized by human expert, who would be able to notice aforementioned positional properties of terms.

It is possible also that long n-gram based representations could be effective here, but their practical usefulness remains limited due to their size, while using short n-gram sequences does not seem to be a very effective method for improving categorization quality [10].

We propose therefore a different approach to this problem, by simply incorporating word position information into document representation.

2 Positional representation

The *positional representation* is a simple extension of a classic bag-of-words representation, which stores not only information about word occurrence frequency, but also information about relative positions of words in a document.

Positional representation of a document $D=(w_1, w_2, ..., z_1, ..., w_n, z_m)$ is a pair $(\boldsymbol{F}, \boldsymbol{S})$ where $\boldsymbol{F}$ is a set of word density functions f_{Vi} such that their domain is a set $\{1..n\}$ and values are defined as follows:

$$f_{v_i}(k) = \frac{\sum_{j=k-r}^{k+r} \begin{cases} 1 \; if \; w_j = v_i, v_i \in V \\ 0 \; otherwise \end{cases}}{\alpha_i} \quad \text{and} \sum_{1}^{n} f_{v_i} = 1 \tag{3}$$

while ***S*** is a scaling vector with same values as in *unigram* representations.

The r parameter may be interpreted as representation fuziness. Note that if *r=n* representation degenerates to simple unigram - *f* is constant and equal *1/n*. On the other hand when *r=0*, representation allows for exact reconstruction of source document.

The density function may be also interpreted as a probability distribution which values reflect the probability of encountering given keyword in specific portions of the document.

Of course using this representation in its direct form is not practical, due to its size. We can howerver replace the density function with approximating histogram which parameters would be stored in a matrix ***M***. The subsequent rows of this matrix correspond to words from system vocabulary, the columns correspond to fragments $(w_a,...w_{a+k})$ of document *D*, and cell values are defined as follows $M_{x,y} = \sum_{i=a}^{a+k} f_{v_x}(i)$ and of course $\sum M_{x,y} = 1$. It is then possible to process such matrix with standard classification algorithm either directly - for example by computing distance between documents as average of dot products of individual columns of their matrices, or indirectly - by deriving source functions for documents and comparing them, what can be done for example by means of functions such as Kullback-Leiber divergence.

Building positional representation for a document is not a complex task, and involves using a window of length *2r* „sliding" over document's contents, counting occurences of words contained within the window, and normalizing count values afterwards (see Figure 4).

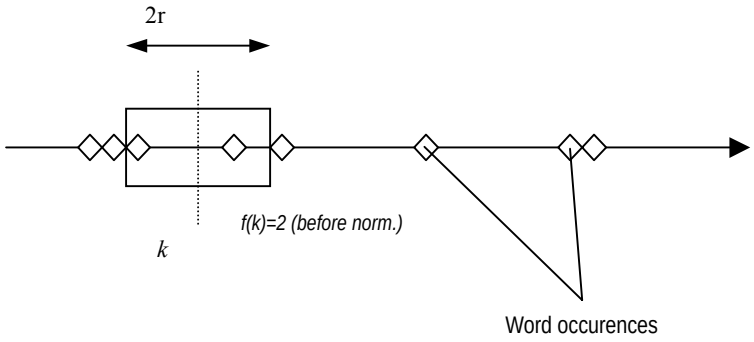

Fig. 4. Creating positional representation

Below we present examples of word density functions for words discussed in previous section, created with two different fuzziness values.

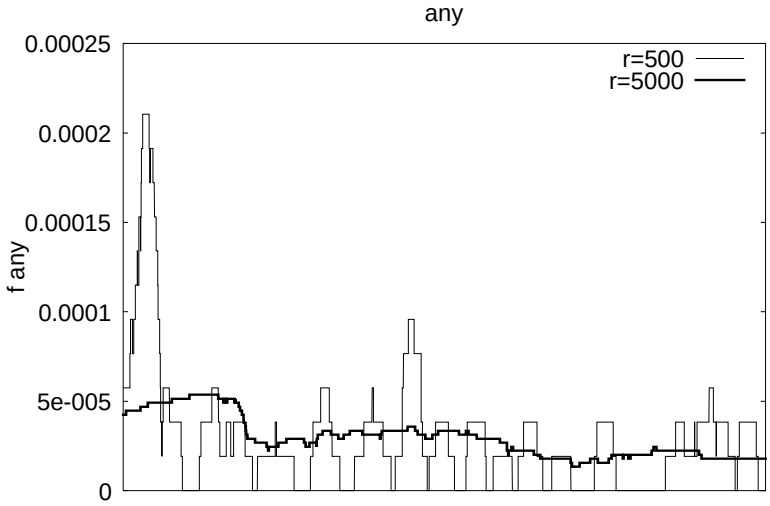

Fig. 5. Density functions for word any in [14]

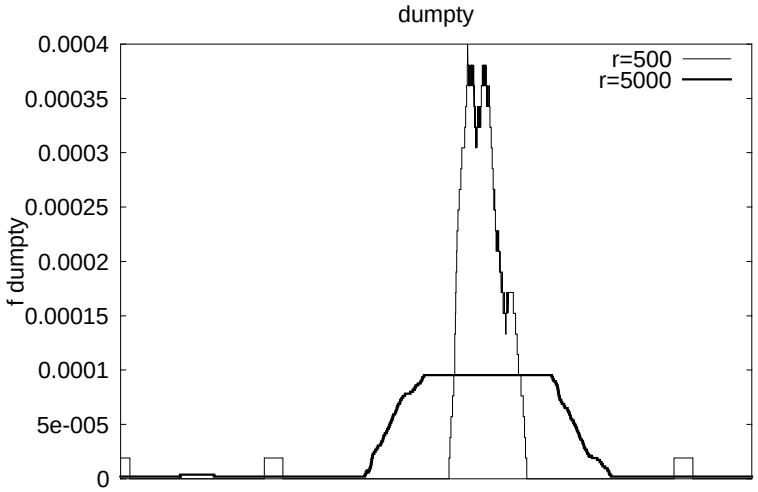

Fig. 6. Density functions for word dumpty in [14]

3. Representation processing

Because the positional representation incorporates normal unigram (represented by scaling vector S) it is possible to apply standard representation processing techniques, such as scaling with *tf-idf* functions, attribute selection and so on. However the infor-

mation about word positions creates new possibilities of analyzing word importance. First of all, individual density functions can be checked for uniformity. Frequent words, randomly scattered throughout the document are likely to be less important, than these concentrated around specific place in it, and hence could be possibly removed from representation. A slightly modified Lorentz concentration function can be used for this test:

$$K_i = 2 \cdot \frac{\sum_{a=0}^{1} \left[a \cdot n - \left(\sum_{x=0}^{n} \min(f_{vin}(x), a) \right) \cdot \frac{n}{\sum_{x=0}^{n} f_{vin}(x)} \right]}{n} \quad (4)$$

where f_{vin} is normalized density function for word w_i such that $max(f_{vin})=1$.

If $K_i = 0$ the word occurrences are distributed uniformly throughout the document. On the other hand, if the maximum word concentration occurs (when there is only single word occurrence in a document) $K_i=1$. Above tendencies may be even stronger, if we sum all density functions for a given word from all documents in certain class. The regularities in word usage (such as - for example - using certain words in documents introductions) will be enhanced. Of course these measures should not be interpreted directly, because of their dependence on absolute number of word occurrences in a document.

Another processing method may involve clustering of these words, which have similar density functions, and therefore are possibly semantically related. Such semantic clustering could be also performed with the aid of manually constructed semantic networks such as Wordnet, however a disadvantage of this method is its language dependency. For many languages (such as Czech, Polish etc.) it is not easy to obtain a general-purpose thesaurus, of scope and quality comparable with Wordnet. This may be however overcome by using automatic thesaurus construction technique proposed by Pedersen and Shutze [16], which uses second order coocurence between terms.

4. Preliminary results and conclusion

The ideas presented briefly above are still far from being completely elaborated, so we definitely consider this as a work in progress. The experiments that are being conducted are mainly focused on testing the usefulness of the positional representation, and also of other types of non-unigram representation such as n-grams and γ-gram (see for example [2]) for various text mining tasks, with special emphasis on document categorization.

One of the examples of preliminary experiments involving positional representation and document formatting has been illustrated below. In this experiment we are trying to assess whether it is possible to perform effective categorization based only on document formatting, without analyzing textual contents of the pages. We created

small, three class system (typical documents contained within these classes are presented in Figure 7 - PAPERS class contains various scientific papers, class LISTS - lists of terms such as dictionaries or search engine results and finally online newspapers main pages have been stored in PRESS class). We than converted Web documents belonging to these classes by replacing word sequences between XHTML tags with one symbol δ, thus achieving the following sequences of tokens $(t_1,..., \delta,... ,t_o)$, where t_i represent various formatting tags, as documents contents. The classifier, using positional representation with 10 column histogram matrix, was able to achieve, for class PAPERS precision 0.9, recall 0.9; for class LISTS precision 0.64, recall 0.9; and for class PRESS precision 1 and recall 0.6. These results are given here only as examples, as the entire experiment is yet far from being completed.

Fig. 7. Examples of documents used in formatting-based categorization system

Other experiments include assessing positional representation-based classifier on Reuters, Digitrad corpora and also on a specially designed corpus containing Project Gutenberg etexts. It is too early to say anything for certain about the results here. We expect that we should be able to obtain small increase in categorization accuracy, but it is also possible that there will be no such increase - especially over Reuters corpus due to it's aforementioned properties. In preliminary tests we used simple classification algorithms, such as Rocchio, and these do produce better results (average 2-3% increase) than unigram, however we expect more interesting data from SVM system that we are currently developing.

The positional representation presented in this paper is a very simple concept. Its usefulness has not been yet fully tested, and it is still possible that further experiments will show that it is unsuitable for text mining applications. Even thou we believe that further research on more complex document representations is necessary, especially as

rapidly increasing capabilities of computer hardware - especially storage, both memory and disk, make possible considering larger representations (in terms of their memory requirements), that were impractical only a few years ago.

References

[1] Shuetze H., Manning C. “Foundations of Statistical Natural Language Processing“, MIT Press, 1999

[2] Gawrysiak P., “Automatic document categorization”, PhD thesis, Warsaw University of Technology

[3] StatSoft, Inc., “Electronic Statistics Textbook”, Tulsa, 2001, http://www.statsoft.com/textbook/stathome.html.

[4] McCallum A., Nigam K., ”A Comparision of Event Models for Naive Bayes Text Classification”, AAAI-98 Workshop on Learning for Text Categorization, 1998

[5] Dumais S. et al., “A Bayesian Approach To Filtering Junk Email“, Microsoft Research, 1998

[6] Apte C., et al, ”Maximizing Text-Mining Performance”, IEEE Intelligent Systems, July/August 1999, p.3-8, 1999

[7] Joachims T., ”Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, Proceedings of 10th European Conf. On Machine Learning, 1998

[8] Dumais S., “Using SVMs for text categorization“, IEEE Intelligent Systems, July/August 1998, p.21-23, 1998

[9] Yang Y., Liu X., „A re-examination of text categorization methods”, ACM SIGIR Conference on Research and Development in Information Retrieval, New York, 1998

[10] Mladenic D., ”Machine Learning on non-homogeneous, distributed text data”, PhD thesis, Lubijana, 1998

[11] Matwin S., Scott S., ”Text Classification Using WordNet Hypernyms”, Computer Science Dept., University of Ottawa, 1998

[12] Reuters-21578 Test Collection, http://www.research.att.com/~lewis/reuters21578.html

[13] Calvo R., “Classifying Financial News With Neural Networks”, Proc. of the 6th Australasian Document Computing Symposium, Coffs Harbour, Australia, 2001

[14] Caroll L., “Through the Looking Glass”, Millenium Fulcrum Edition 1.7, Project Gutenberg Etext, 1994

[15] Hearst M. , “Context and Structure in Automated Full Text Information Access”, UC Berkeley Computer Science Technology Report UCB:CSD-94-836

[16] Pedersen J., Schuetze H., “A coocurrence-based thesaurus and two applications to information retrieval”, Information Processing and Management, 1995