Maximal Phrases Based Analysis for Prototyping Online Discussion Forums Postings

Gaston Burek Dale Gerdemann
Department of Linguistics
Tuebingen University
72074 Tuebingen, Germany
[gaston.burek, dale.gerdemann]@googlemail.com

Abstract

Chat texts produced in an educational environment are categorized and rated for the purpose of positioning (or placement) of the learner with respect to a learning program (appropriate courses, textbooks, etc). The difficulty lies in the fact that the texts are short and informal. A standard LSA/vector-space model is therefore combined with techniques appropriate for short texts. The approach uses phrases rather than words in the term-document matrix, and for determining prototypical documents of each category, a nonparametric permutation test is used.

1 Introduction

Text categorization is a well-established sub-field of computational linguistics. We are interested in applying the techniques of text categorization for the purpose of positioning *Life Long Learners* with respect to educational programs and instructional materials. Quite simply, we can rate learner texts, likely to be short and generated informal educational settings, by using a vector-space comparison to gold-standard, expert texts. Then if the similarity is high enough, the learner will proficiency out of the course. This approach is straightforward, but is unlikely to be successful. The problems concern accuracy, suitability and justification of the categorization.

1.1 Accuracy

The categorization obviously must be accurate. A false positive, indicating learner proficiency in a particular domain, could be dangerous as it could lead to work place incompetence. A false negative, on the other hand, could lead to boredom, as the learner is forced to take courses on topics that he or she has already mastered.

The problem of accuracy is compounded by the fact that texts (selected from text collections known as *ePortfolios*) are often short. To deal with this problem, our approach attempts to lose as little information from the text as possible. Traditional approaches to categorization lose information by case normalization, stemming and ignoring word order. The idea of the traditional approach is to deal with the data sparseness problem by collapsing textual features into equivalence classes, losing information in the process.

In our approach, we attempt to balance the problem of data sparseness with the goal of not losing information. This balance is obtained in two ways. First, as discussed in section 2, we use *Latent Semantic Analysis* (LSA) as a technique for dimensionality reduction [8]. It is well known that LSA can be used to discover weighted clusters of words, which are loosely understood to be “concepts.” Since these clusters can contain derivationally related terms, the need for stemming (and also case normalization) is reduced. Second, as discussed in 3, our more innovative contribution is to flexibly use bigrams, trigrams and other ngrams as opposed to strictly unigrams in the traditional bag-of-words model. Our approach to extracting such ngrams is to use a extension of the suffix array approach of [12].

1.2 Suitability

Suppose that learner texts could be accurately classified as similar or not similar to the gold standard text (or set of texts). Then the question arises as to whether or not the gold standard text is a suitable prototype for a good learner text. One approach to choosing a gold standard text would be to use a published journal article in the field. But such a text is unlikely to be similar to learner texts either in tone or in content. It is well known that effective teachers use *scaffolding* to present material within the *zone of proximal development* of the learner.

So perhaps a better gold standard would be a textbook, or other learning material, written at the level of the student. This is certainly an improvement, but on the other hand, it is still rather unreasonable to expect learners’ texts to closely match the tone of a textbook, unless of course the learners are copying from the text. In fact, the texts that we have consist of online discussions of medical students on several topics related to safe prescribing. These texts have been categorized as to subtopic and graded for quality (excellent, good, fair, poor) by annotaters at the University of Manchester. The texts contain serious conversations, with very little off-topic wandering. But the tone of the texts is chatty, and not at all similar to textbook writing. So rather than to use an external gold standard, we have opted for an internal gold standard. The prototypical “excellent” text is simply one that was rated as “excellent” by the Manchester annotators. But not all “excellent” texts are equally good as prototypes. Clearly, for any text t, the remaining texts can be ranked in order of similarity to t. If t is a good prototype, then this ranking should other “excellent” texts as most similar to t and “poor” should be least similar. In subsection 4, we discuss a method for choosing good prototypes that uses the nonparametric *permutation test.*

1.3 Justification: communities of practice distinctive language

It is widely accepted that experts can provide answers to problems that average people can not e.g. evidence given in court can be accepted or rejected on the basis of expertise, relevance and reliability. In this contexts expertise is defined as knowledge beyond common sense [6].

Communities of practice are at the center of expert knowledge development. According to [4] new communities of practice develop conditions for effective local creation and communication of knowledge and therefore that knowledge is not shared with other communities until it settles. In linguistic terms this can be understood as a distinctive use of language shared among individuals members of a specific community to describe knowledge that is not shared with other communities of practice. According to [2] expert language in medicine consists in shared specific terms formalized in medical term banks, thesauri and lexicons. Patients get familiar with that established medical terminology relevant to their own health conditions medical language after exposure to treatments, personal doctor visits, etc.

1.3.1 Language technologies supporting the identification of expertise

As life long learners do not have common learning goals nor common educational backgrounds as it is the case in traditional learning settings, long life learning educational providers need to rely on available written materials produced by individual learner to identify their degree of expertise within areas of knowledge that are relevant to study programs in offer.

Given this scenario there is a need for tools that can provide support in determining the learner’s degree of expertise or position by means of state of the art language technologies. Those technologies should be capable of identify linguistics features that reflect the degree of learner’s expertise by analyzing learners’ text repositories.

But, will learners affected by the use of such technologies (e.g. LSA) be happy to hear about a expert system suggestion that implies the need of studying a microbiology course on account of a low cosine similarity value between the learners essays and texts produced by microbiology experts? It is well known that users are more inclined to trust an expert system when such tool can give some reasons for its judgement.

Here, in addition to use LSA for comparing similarities between learner written text and expert texts, we present the use suffix array analysis for characterizing text in a way that users of that technologies can understand .

Grammar and language constructions can be used to identify language that is characteristic of people that is not familiarized with a relevant communities of practice e.g. microbiology . This linguistic feedback could be combined with a tentative suggestion that taking a microbiology course would be one way to acquire these linguistic conventions that tend to correlate with knowledge of microbiology.

Our work focus on the use analysis of distinctive phrases for identifying the degree of expertise of it author in a specific domain where that expertise may be expressed by linguistics features that is not be evident at surface level. This approach determine that degree of expertise as the probability of language misuse in relation to average language use as sampled from expert written texts.

To give an example from and available, consider the terms ”prescription charts” and ”drug charts.” These terms apparently mean the same thing, but it turns out that ”prescription charts” occurs predominately in texts rated excellent or good, and ”drug charts” occurs predominately in texts rated fair or poor. Suppose that the hypothesis that doctors either consciously or unconsciously prefer the term ”prescription charts.” Then if a learner uses the term ”drug charts,” this usage could be tagged as less favored terminology, and a tentative suggestion could possibly be made that the learner could take a pharmacology course. If the learner ultimately rejects this suggestion, then at least this learner wouldn’t go home empty handed. The learner would at least have received some linguistic feedback that would be unlikely to come from a human evaluator.

2 Latent Semantic Analysis

In recent years, LSA has been proposed as a suitable language technology for the automatic determining the degree of expertise of a specific text’s author [10]. Although, singular value decomposition (SVD) of word co-occurrence frequencies matrices has been successfully used in the context of language technologies enhanced learning (e.g. automatic assessment of student essays), learner positioning presents new challenges that expose the limitation of such an approach. In particular, learners produce text repositories containing few samples of text, many of them of small size and using language that is rich in non domain-specific expressions. Such repositories may also be generated by individuals from different backgrounds in informal learning environments (chats, online forums, etc.). In addition, those texts are generated in contexts where learners feel encouraged to hide their poor usage of language by articulating redundant expressions and making extensive use keywords. Moreover, linking the semantics of high level descriptions of learning goals and domain specific terminology used by learners in a non-formal context restrict the usability of LSA as word usage may not be very stable across corpora.

While semantic spaces approaches such as the Vector Space Model (VSM) captures surface features of a semantic space such as first or second order word co-occurrence, the semantic similarity theory behind LSA is based on a model that captures different higher orders of word co-occurrence (i.e. third order co-occurrence and higher) by means of using singular value decomposition. LSA uses SVD to project the semantic space over a lower dimensional space. LSA supporters claim that by reducing the space dimensionality, the semantic space that models the human cognitive process becomes more accurate due to the reduction of the noise that is added during the process of generating language.

LSA belongs to the family of semantic space models and is capable of inferring relations between words that are not explicit at surface level. LSA is popular with psychologists since the dimensionality reduction can be interpreted as reducing the word space to concept space. But ultimately these "concepts" are loadings on word counts, which are not easily interpreted or explained.

LSA as well the other semantic spaces approaches represents the semantics of words, sentences and paragraphs using word co-occurrence information that does not take in consideration the position of words within the sentence. However, according to a general conception of semantics, syntax plays a significant role in representing meaning of sentences. Thus, intuitively to using information about the order words occupy within expressions in addition to their frequency of occurrence seems to be a theoretically sounded approach to improve LSA performance that is in line with mayor trends in language acquisition theories that stress the significance that syntactic structure play in the comprehension of language.

Although, semantic spaces approaches (e.g. Latent Semantic Analysis) have been successful in reasoning about ambiguity and semantic similarity when analysing texts at the level of words (linguistic units), sentences (grammatical units) and paragraphs (discourse units) they are yet not capable of reasoning beyond the analysis of a bag of words.

Typically, LSA projects a semantic space implemented as a sparse matrix into a dimensionally reduced VSM generating the best statistical approximation to the original model. The dimensionally reduced model results in a matrix with non 0 values that allows the computation of similarity between matrix columns or rows that were orthogonal in the original model.

LSA theory assumes that many words have a similar probabilistic distribution that results in a comparatively lower number of concepts.

As the variation of word choice, introduce noise to the text, SVD, the algorithm behind LSA bridges the gaps in words by conflating word used into word senses . Furthermore, LSA facilitates matrix manipulation in terms of hardware operative memory by reducing the dimensionality of the VSM.

3 Phrases

Traditionally, text categorization by means of LSA has relied on a bag-of-words model. It seems, in some sense, obvious that a model based on phrases should be better. But it turns out that this is not necessarily the case. Recently, Bekkerman & Allan [3] reviewed the literature on text categorization and found no general improvement when unigram models were replaced with bigram models. The problem is that using bigrams contributes heavily to the data sparseness problem.

Bekkerman & Allan have, however, compared two rather extreme positions. Our idea is to extract phrases of any length from the the training corpus, as long as the phrases are distinctive (occurring predominately in particular categories of documents). It may well be that the most distinctive phrases are generally phrases of length one (concurring with the bag-of-words model), but if there are phrases of other lengths that are more distinctive, then there seems to be no reason not to use these phrases.

To give an idea of the approach, consider the word *side*. In the medical discussions in our corpus, this word almost always occurs as part of the phrase *side effect(s)*. In a few cases, *side* occurs in a unique context or as part of another phrase, such as *flip side*. In this case, the distinctive phrase is apparently *side effect*, and the other occurrences are just noise. These noise phrases are not only unhelpful for text categorization, they are are also unhelpful for generating explanations that would be useful for learners and examiners.

The example above raises some interesting counting issues. But first we need to specify more precisely what it means for a phrase to be distinctive.

3.1 Distinctiveness

In general, phrases that are evenly distributed across document categories are not very distinctive, whereas phrases that tend to cluster in one particular category are distinctive. This general principle must be applied carefully, however, since with small numbers, clustering may occur due to chance.

A common measure of distinctiveness used for weighting in vector space models is *tf-idf* (term frequency multiplied by inverse document frequency) [9]. It is unclear, however, that this is the best measure for picking out which phrases to consider and which phrases to ignore. It is problematic, for example, that *idf* simply prefers terms that cluster in a small number of documents, regardless of the classifications. Given the ordinal classification of our data as *excellent*, *good*, *fair* and *poor*, we are not interested, for example, in terms that cluster in the *excellent* and *poor* texts.

So a distinctive term should be one that occurs predominately in *excellent* and *good* texts or predominately in *fair* and *poor* texts. Consider, for example, the bullet point , with occurrence vector $\langle 31, 5, 0, 0\rangle$.[1] The interpretation is that there are 31 occurrences in *excellent* documents, 5 occurrences in *good* documents and no occurrences in either of the poorer texts, this term appears be very distinctive of better texts. But if we count instead the number of different documents the bullet point occurs $\langle 4, 1, 0, 0\rangle$, we see a very different picture. The bullet point does occur in higher rated texts, but it is very bursty (cite Church) and is therefore not very useful for categorization.

There are various approaches in the literature for dealing with burstiness. Since this is not our primary

[1] Optionally the tokenizer could be set to eliminate such punctuation marks. The bullet point makes a good example here, however, due to its burstiness.

concern here, we deal with the problem by counting the number of texts containing a term rather than the total number of occurrences of the term. Thus, for the bullet point, we use the vector $\langle 4, 1, 0, 0\rangle$.

To rate a term such as the bullet point, we need some measure of goodness for the vector $\langle 4, 1, 0, 0\rangle$. There is clearly no objective measure that can be used here. As a fairly reasonable score, we simply assign 1 point for every *excellent* text, 0.8 points for every *good* text and 0.2 points for every *fair* text. So, the bullet point receives a score of 4.8. This appears to be a good score, but what is the probability that a randomly chosen term appearing in 5 texts would have a higher or equally high score? We can generate random vectors as in (3.1) (where e, g, f and p are the total numbers of *excellent*, *good*, *fair* and *poor* texts, respectively.

$$X_i = \begin{cases} \langle 1, 0, 0, 0\rangle & \text{with probability } \frac{e}{e+g+f+p} \\ \langle 0, 1, 0, 0\rangle & \text{with probability } \frac{g}{e+g+f+p} \\ \langle 0, 0, 1, 0\rangle & \text{with probability } \frac{f}{e+g+f+p} \\ \langle 0, 0, 0, 1\rangle & \text{with probability } \frac{p}{e+g+f+p} \end{cases}$$

A vector for a random term occurring in n texts is then $\sum_{i=0}^{n} X_i$. So for a good score such as 4.8, the idea is to see what proportion of randomly generated vectors have an equally high or higher score. And for a low score, the opposite idea is to count the proportion of randomly generated scores that are equal or lower.

3.2 Phrase Extraction

In principle, the distinctness measure given above can be used with phrases of any length. If longer phrases can be found that are more distinct than single words, then there is no reason not to use the longer phrase. The problem is that the simulation-based distinctness test is very expensive, and it is certainly not possible to run this test for ngrams of every length in a text. The solution to this problem comes from Yamamoto & Church [12], who show suffix arrays can be used to put the large number of ngrams into a much smaller number of equivalence classes. Using suffix arrays, it is very easy to pick out just the phrases that are repeated n times for some n, and it is very easy to extend phrases to the right: if *mumbo jumbo* repeatedly occurs together as a phrase, then it makes no sense to count *mumbo* by itself. Yamamoto & Church's suffix array program will put these two phrases into an equivalence class, so that that statistics can be calculated for the class as a whole rather than individually for all the members of the class.

Since the time of Yamamoto & Church's paper, suffix arrays have been an active area of research, primerily in bioinformatics. One of the weaknesses of the suffix array approach used by Yamamoto & Church is that extensions to the left are difficult to discover. So it is difficult to discover, for example, that *jumbo* always combines to the left to form the phrase *mumbo jumbo*. Simply stated, the problem is that suffixes are extensions of phrases to the right, so it is hard to look to the left. This problem was solved, however, by Abouelhoda et al [1], who added a BurrowsWheeler transform table to their extended suffix array data structure, giving this this data structure properties of suffix trees.

One weakness of Abouelhoda et al's approach, however, is that it does not adapt well to large alphabets. This is, of course, a serious weakness for use in text processing, where one wants at least to work with some subset of Unicode, or even worse, to treat each tokenized word as an alphabet symbol. Fortunately, the restriction to small alphabet size has recently been eliminated in the approach of Kim et al [7], who deal with the large alphabet by using binary trees, which are linearly encoded using the *child table* of Abouelhoda et al along with a *longest common prefix* table (lcp).

Using extended suffix arrays makes it possible to count different kinds of occurrences of phrases in different ways. To begin with, we are only interested in counting phrases that repeat. In the text $S =$ *to be or not to be*, the occurrence of the phrase *to be* at $S[1, 2]$ is said to be a *repeat* since the same sequence of tokens occurs at $S[5, 6]$.[2] An occurrence of a phrase $S[i, j]$ is *left maximal* is *left maximal* if the longer phrase $S[i-1, j]$ is not a *repeat.* Thus, for example, the phrase *to* at $S[1, 1]$ is *left maximal* since the phrase at $S[0, 1]$ is not a *repeat.*[3] Similarly, an occurrence of a phrase at $S[i, j]$ is right maximal if $S[i, j+1]$ is not a repeat. If an occurrence of a phrase is both left and right maximal, then the occurrence is said to be *maximal.* Note that the occurrence of the phrase *or not* at $S[3, 4]$ is *maximal*, though it is not a repeat. Since non-repeats are rarely of interest, we generally assume that we are talking about repeats unless otherwise stated.

A phrase is also said to be *maximal* in a text if there exists a maximal occurrence of the phrase in the text. For example, in the text *mining engineering*, tokenized by characters, the phrase *in* is maximal since there are maximal occurrences at $S[2, 3]$ and $S[11, 12]$. But the longer phrase *ing* is also maximal since it occurs maximally at $S[4, 6]$ and $S[16, 18]$. So the occurrence of *in* at $S[16, 17]$ is a non-maximal occurrence of a maximal phrase. A maximal repeated phrase that is not a subsequence of a longer maximal repeated phrase is said to be *supermaximal.* Thus the phrase *ing* is supermaximal in this text.

Generally, we are only interested in counting occurrences of maximal phrases since a phrase that never occurs maximally is unlikely to be of interest. But what kind of occurrences should we count? Should we count all occurrences, or only the left maximal, right maximal or maximal occurrences? The answer is that we don't need to decide ahead of time. We can simply test each of these four cases for distinctness, and chose the most distinct case. Take, for example the word *side*, which is a maximal phrase in our texts. Should we count all instances of this phrase? Or should we perhaps restrict the count to right maximal occurrences so as to avoid counting those instances that are extended to the right to create the longer phrase *side effect*? Or maybe left maximal occurrences to avoid the longer phrase *flip side*? Or perhaps we should restrict

[2] This definition and the following definitions are similar to those found in Abouelhoda et al [1]. The difference is that Abouelhoda et al apply the terms to a pair of occurrences, whereas we apply the terms to a single occurrence.

[3] We assume here that the text is padded with unique *beginning of string* and *end of string* sentinels so that indexing at 0 or 7 makes sense.

in both directions to avoid either kind of extension. Since it is not generally possible to predict which is best, the reasonable approach is to try all possibilities to see what works best.

One counterintuitive feature of our approach is that it also makes sense to count 0-grams. A left maximal occurrence of a 0-gram, for example, must have a hapax legomena to its left, and a maximal occurrence of a 0-gram must have hapax legomena on both sides. These sequences of two hapax legomena may well be distinctive, since they often are an indication of a named entity or a foreign phrases. Counting all occurrences of the empty sequence is, of course, equivalent to counting the text length, which may well also be a distinctive feature.

4 Permutation test for prototyping chat texts

Our approach to categorization is based on similarity to prototypical documents of each category. Here again, we are concerned with finding an approach of identifying good prototypes in a way that is suitable for use with a small data set. We note that each candidate prototype induces a ranking of the remaining documents from "most similar" to "least similar." For a good prototype, this ranking should be significantly different from a random permutation. To test this hypothesis, we can use a standard nonparametric test, known (appropriately enough) as the "permutation test." The permutation test is a nonparametric method for testing whether two distributions are the same. The test is "exact," meaning that it is not based on large sample theory approximations [11]. Suppose that $X_1, ..., X_m \sim F_X$ and $Y_1, ..., Y_n \sim F_Y$ are two independent samples and H_0 is the hypothesis that the two samples are identically distributed. This is the type of hypothesis we would consider when testing whether a treatment differs from a placebo. More precisely we are testing $H_0 : F_X = F_Y$ versus $H_1 : F_X \neq F_Y$. Let $T(X_1 ..., X_m, Y_1, ..., Y_n)$ $=|\overline{X}_n - \overline{Y}_n|$ and let $N = m + n$ and consider forming all $N!$ permutations of the data $X_1 ..., X_m, Y_1, ..., Y_n$. For each permutation, compute the test statistic T. Denote these values by $T_1, ..., T_{N!}$. Under the null hypothesis, each of these values is equally likely. Let t_{obs} be the observed value of the test statistic. We reject the hypothesis when T is large.

Usually, it is not practical to evaluate all $N!$ permutatioons. We can approximate the $p - value$ by sampling randomly from the set of permutations. The fraction of times $T_j > t_{obs}$ among these samples approximates the $p - value$. In general If T is smaller than some significance level a , the results are significant at that level. In our case we are not concern about a particular significant level as we are looking to the most significant texts on certain topic and with certain grade that can be used use as gold standards. Table 1 shows a toy example for the cosine similarity vector $(X_1, X_2, Y_1) = (0.1, 0.5, 0.8)$ where, $|\overline{X}_n - \overline{Y}_n| = 0.50$.

The methodology described in this section can be used as a reasonable approach for measuring how significant (representative) chat texts are for a particular category and grade. The estimation is calculated on the basis of cosine similarity between the texts and does not assume any particular distribution for those values.

permutation	value of T	probability
(0.1, 0.5, 0.8)	0.50	1/6
(0.1, 0.8, 0.5)	0.05	1/6
(0.5, 0.1, 0.8)	0.50	1/6
(0.5, 0.8, 0.1)	0.55	1/6
(0.8, 0.1, 0.5)	0.05	1/6
(0.8, 0.5, 0.1)	0.55	1/6

Table 1: *Permutation example*

5 Experiments

The experiments described in this section compare the performance of the traditional bag of words LSA configuration against an alternative configuration that uses maximal phrases as unit of analysis.

The alternative LSA configuration starts with a vector space model that (instead of using words counts) uses counts of highly distinctive phrases that occur at least one time as maximal phrase within the text collection under analysis.

The use of a "bag of phrases" model instead of a "bag of words" model is motivated by the small size of our sample annotated of chat texts. Since our information source is scarce, we cannot afford lose any of it. If a medical text contains the phrase "side effect," for example, it is significant information that these two words occur as a phrase, which would be lost in the traditional bag-of-words approach.

Our ongoing work in word co-occurrence models for learner positioning extends existent LSA based approaches and is aimed at analyzing and then scoring texts posted on an online medical student discussion forum where students discuss issues related to the safe prescribing of medicines, guided by clearly defined expected learning outcomes associated to six subtopics. The 504 postings have been annotated by experts with four grades (i.e. 109 poor, 200 fair, 142 good, 50 excellent) and one of six topics (i.e. 42 of topic a , 50 of b, 130 of c, 22 of d , 247 of e and 13 of f). Each grade is based on the individual posting's textual contribution to a series of expected learning outcomes. Highly scored postings can then be used as evidence of learner proficiency in the corresponding topic.

5.1 Building the bag of words and phrases based semantic spaces

As explained in 3.2 to identify and extract the maximal phrases we analyze suffix arrays using an extended version of the the Yamamoto and Church algorithm to generate all ngrams from a text and avoiding the combinatorial explosion by grouping these ngrams into equivalence classes. This approach, however, has the disadvantage of generating also "uninteresting" ngrams, which are neither maximal nor supermaximal. To overcome this deficiency, we employ the Burrows and Wheeler transformation table.

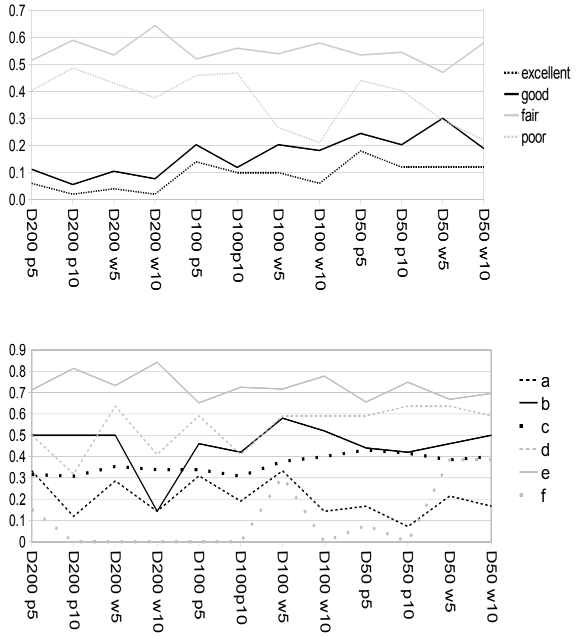

Fig. 1: *kNN results for neighborhood sizes 5 and 10 and semantic spaces built from phrases (p) and bags of words (w) using 5, 100 and 200 singular values*

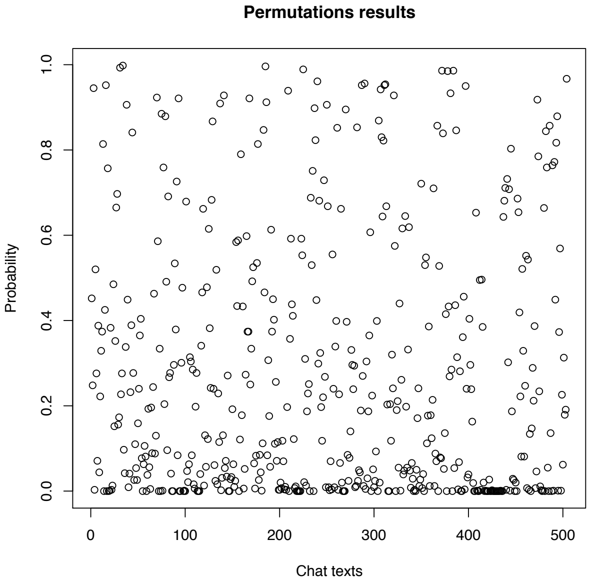

Fig. 2: *Permutation test results for each chat text*

Each phrase has been counted in one of 4 ways: all instances, left-maximal, right-maximal and maximal. To avoid an unmanageable level of sparseness we include in the analysis all instances of all phrases that occurs at least one time as maximal. Phrases are sorted by their scores absolute values.

We then built a 19730 phrases to 504 chat texts matrix that contains the frequency of occurrence of each phrase in each texts. We then weighted the matrix using the $tf - idf$ weighting scheme. We then generate three LSA semantics spaces by reducing the SVD resulting matrix singular values to 50, 100, and 200 respectively.

In addition we created another set of 3 bag of words based LSA semantic spaces using the same weighting scheme and respective number singular values. In this case using a 6320 tokens to 504 chat texts matrix. The number of token used is the results of choosing the tokens that occurs at least two times within the chat texts collection.

5.2 k Nearest Neighbor algorithm based classification

The k Nearest Neighbors algorithm (kNN) [5] is a learning algorithm that classifies texts on the basis of a measure of distance (e.g. cosine) between them. The algorithm classifies each text by looking at a k number of its nearest neighbors and then assigning it to the most common category represented by those neighbors. If no class is associated to a majority of neighbors, the text is assigned to the category represented by texts with higher cosine similarity. We arbitrarily used a low k value (i.e. k=5) as we expect that noise from the semantic space will be reduced by means of LSA. A common criticism of kNN is that, since it doesn't make any generalizations from the data, it is prone to overfitting. We assume that this criticism should not apply completely to semantic spaces generated by means of LSA as the SVD dimensional reduction smoothes and therefore reduces the effect of over fitting that is usually present in kNN based classification.

5.2.1 kNN results

Experimental results showed in Figure 1 demonstrate that for some topics and grades using maximal phrases as units of analysis can improve the performance of LSA. In fact the best results for two of the four grades (e.g. excellent and poor) were yielded by semantic spaces build from phrases. Different results are produced by kNN classification for topics where semantic spaces built from bags of words produced the best results clearly for at least four of the six topics.

5.3 Prototyping

We run the permutation test described in section 4 using the already available semantic space built from a 19730 phrases to 504 chat texts matrix and singular values reduced to 200. We then evaluate the significance of each text for its annotated grade and topic category. Figure 2 shows that the majority of texts have low probability as the majority of them may not be a good representative of their class.

6 Conclusions

For particular grades and topics phrase based LSA (i.e. using semantic spaces built from phrases occurring at least one time as maximal) appears to improve over LSA results that have been obtained with the traditional bags of words approach. These results are encouraging and therefore we plan to test alternative semantic space configurations in particular using more distinctive phrases (e.g. all maximal, left maximal and right maximal). We expect that as we collect a larger text sample we will be able to afford the use of those phrases without facing unmanageable levels of sparseness in detrimental of results already obtained. In addition, we want to stress the fact that the suffix array analysis presented in this paper is independent from its LSA application as it can be used to characterize learners and experts use of language. In addition, as our approach to categorization is based on similarity to prototypical texts for each class we presented here an non parametric test (permutation test) for identifying those prototypes.

References

[1] M. I. Abouelhoda, S. Kurtz, and E. Ohlebusch. Replacing suffix trees with enhanced suffix arrays. *J. of Discrete Algorithms*, 2(1):53–86, 2004.

[2] H. Åhlfeldt, L. Borin, P. Daumke, N. Grabar, C. Hallett, D. Hardcastle, D. Kokkinakis, C. Mancini, K. Markó, M. Merkel, C. Pietsch, R. Power, D. Scott, A. Silvervarg, M. T. Gronostaj, S. Williams, and A. Willis. Literature review on patient-friendly documentation systems. Technical Report 2006/04, Centre for Research in Computing, The Open University, Milton Keynes, UK, May 2006. ISSN: 1744-1986.

[3] R. Bekkerman and J. Allan. Using bigrams in text categorization. Technical Report IR-408, Center of Intelligent Information Retrieval, UMass Amherst, 2004.

[4] J. S. Brown and P. Duguid. Knowledge and organization: A social-practice perspective. *Organization Science*, 12(2):198–213, March 2001.

[5] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[6] A. Davies and C. Elder, editors. *The Handbook of Applied Linguistics*. Blackwell Handbooks in Linguistics. Wiley-Blackwell, Malden, 2006.

[7] D. K. Kim, M. Kim, and H. Park. Linearized suffix tree: an efficient index data structure with the capabilities of suffix trees and suffix arrays. *Algorithmica*, 52(3):350–377, 2008.

[8] T. K. Landauer and S. T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240, April 1997.

[9] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. In *Information Processing and Management*, pages 513–523, 1988.

[10] J. van Bruggen, P. Sloep, P. van Rosmalen, F. Brouns, H. Vogten, R. Koper, and C. Tattersall. Latent semantic analysis as a tool for learner positioning in learning networks for lifelong learning. In *British Journal of Educational Technology*, number 6, pages 729–738, 2004.

[11] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer, September 2004.

[12] M. Yamamoto and K. W. Church. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. *Comput. Linguist.*, 27(1):1–30, 2001.