# Integrating User Preferences Into Distance Metrics

**Mark Kröll**
Know-Center Graz, Austria
mkroell@know-center.at

**Vedran Sabol**
Know-Center Graz, Austria
vsabol@know-center.at

**Roman Kern**
Graz University of Technology, Austria
rkern@tugraz.at

**Michael Granitzer**
University of Passau, Germany
michael.granitzer@uni-passau.de

## Abstract

Many practical scenarios such as search, classification or clustering benefit from better understanding their users, for instance, to deliver more relevant search results. Instead of committing ourselves to a specific field of research, e.g. by generating user profiles to enhance information retrieval, we seek to incorporate user preferences into the distance metric itself which lies at the heart of many algorithms including Information Retrieval and Machine Learning. The two approaches we explore in this paper allow users to directly convey their preferences in an intuitive way. The first approach adheres to the idea that just stating whether two documents are similar or not is more intuitive for a user than, for instance, assigning them to a broad spectrum of topics. The second approach seeks to take into account a user's mental construct of the world being provided with a user-specific concept hierarchy. To evaluate our two approaches, we perform a text classification task. In the classification setting we use the Reuters RCV1 corpus to simulate user preferences. Our results indicate the principal feasibility of these two approaches and encourage further investigations.

## 1 Introduction

Calculating the similarity between textual resources lies at the heart of many algorithms including Information Retrieval, Text Mining or Machine Learning algorithms. Traditional approaches such as TF-IDF [Salton and McGill, 1986] often apply weighting schemes to adapt the impact of certain terms. Yet, a drawback these parametric approaches suffer from is that they are not capable of taking into account user interests.

Practical scenarios such as *search* benefit from better understanding their users. To provide more relevant documents, information retrieval applications aim to personalize search results, e.g. by integrating user interests (cf. [Qiu and Cho, 2006]) or by actively learning search result rankings (cf. [Radlinski and Joachims, 2007]). Other approaches choose a more direct approach by allowing user interaction to convey their preferences. In that sense, users are often required to tune parameters, e.g. decide on cluster size or on the number of neighbors, which affect an algorithm's internal functionality. Yet, adapting these parameters might be counter-intuitive or might require expert knowledge in the sense of a deeper understanding of the algorithm.

We therefore seek to incorporate user preferences into the similarity calculation in a more intuitive manner. Our first approach adheres to the idea that just stating whether two documents are similar or not is more intuitive for a user than, for instance, assigning them to topics (cf. [Saaty, 2008]). In psychology, the idea of using paired comparisons to gain ranking information is a long-established one (cf. [Thurstone, 1927]). In a second approach we seek our distance metric to reflect a user's mental construct of the world by exploiting information from a user-specific concept hierarchy. In this paper, we raise awareness of intuitively incorporating user preferences into the computation of document similarity. In addition, we provide implementations of these two approaches and discuss their characteristics as well as lessons learnt. Finally, we evaluate them in a practical application scenario, i.e. text classification.

## 2 Related Work

In the following, we review work from two fields of research, (i) semantic representation of textual resources and (ii) learning semantic similarity metrics for textual resources.

### 2.1 Semantic Representation

Introducing semantic similarity between features often refers to introducing dependencies amongst formerly unrelated feature dimensions. Attempts to incorporate semantic knowledge into the classical vector space representations include semantic networks, latent semantic indexing or co-occurrence analysis where a semantic relation is assumed between terms whose occurrence patterns in the documents of a corpus are correlated [Cristianini *et al.*, 2002]. Especially kernel-based methods represent an attractive choice for inferring relations from textual documents since they enable a document-by-document setting rather than a term-by-term setting. [Basili *et al.*, 2005] accessed WordNet as external lexical knowledge base to include semantics into the description of textual resources. In their setting they analysed the performance of small-sized training sets for the task of text classification. External knowledge was also used by [Gabrilovich and Markovitch, 2007] which represented the meaning of texts in a high-dimensional space of concepts derived from Wikipedia .

## 2.2 Learning Semantic Similarity

Parametric approaches suffer from the drawback that they do not adapt to particular domains or do not take into account users' personal requirements. [Metzler and Zaragoza, 2009] overcame the rigidity of parametric weighting schemes by introducing semi-parametric and non-parametric weighting schemes. In supervised learning settings, for instance, nearest neighbor classification (cf. [Weinberger and Saul, 2009]), numerous attempts have been made to define or learn either local or global metrics for classification. A number of researchers have demonstrated that nearest neighbor classification can be greatly improved by learning an appropriate distance metric from labeled examples. [Shalev-Shwartz *et al.*, 2004], for instance, optimized the Mahalanobis distance via linear transformations in order to boost the accuracy of a k-NN classification algorithm, which can be seen as implicit application of a weighting scheme.

# 3 User Preference Integration

The integration of user preferences into the similarity computation can be regarded as some form of semantic enrichment. In that sense, semantically enriching the documents' content allows influencing their similarity by introducing dependencies amongst formerly unrelated feature dimensions, as for instance a semantic kernel does (cf. [Cristianini *et al.*, 2002]). We explore two approaches to incorporate user preferences in a more intuitive way and describe implementation details, i.e. how we accordingly adapt underlying distance metrics. For evaluation purposes, we perform a text classification task, i.e. classifying documents from the Reuters RCV1 corpus, a well-known benchmark dataset. In both approaches we use Reuters RCV1 document-to-topic mapping to simulate user preferences.

## 3.1 Similar Document Pairs (SDP)

To adhere to the idea of stating whether two documents are similar or not, we process and merge document pairs to generate new samples. A positive sample is formed by two documents belonging to same category; a negative one by taking two documents belonging to different categories. In our experiments we use a component-wise multiplication (Hadamard product) which results in strengthening common dimensions.

In a first step, the documents' input space is transformed into a higher dimensional space by including bigrams and named entity information, i.e. a concatenation of several feature types. To generate a new sample, we merge two documents by performing a component-wise multiplication. This multiplication results in a new sample vector exhibiting the same dimensionality. In the training phase, we perform an offline processing of the Reuters RCV1 corpus and store relevant information in Lucene[1] indices for fast feature engineering. We then generate new training/test data splits by merging pairs of documents (Hadamard multiplication). From preliminary experiments we learnt that some sort of "intelligent sampling" is required, i.e. "sampling" to keep the number of training/test data manageable in the optimization step and "intelligent" to choose appropriate negative examples. From a class distribution point of view, these negative examples lie close to the boundary of the positive class. To perform this intelligent sampling, we utilize Lucene's search functionality. For every selected document, we search the index for the top $n$ most similar samples once bearing the same class label and once bearing a different class label. These samples are considered for the merging procedure. We then apply Vowpal Wabbit[2], an optimization toolkit, to learn the importance of feature dimensions, i.e. to learn regression weights which are optimized with respect to the new binary classification problem. We remark that in this setting the prior multi-class classification problem is transformed into a binary one. The testing phase handles previously unseen data items, i.e. generating feature types on the fly to calculate similarity values. The same processing steps have to be applied, i.e. multiplying two input documents to determine whether they are similar or dissimilar. Lastly, this new vector is then "informed" by the learnt weights.

## 3.2 Personal Concept Hierarchy (PCH)

In this approach the user provides us with her personal concept hierarchy whose semantic concepts are representative for a certain domain. We then map documents onto these semantic concepts and then apply standard similarity metrics such as cosine distance. Semantic concepts may correspond to categories in a taxonomy as for instance in our case to Reuters RCV1 topics or to Wikipedia concepts as it is done in [Gabrilovich and Markovitch, 2007].

In the training phase, we perform an offline processing of the Reuters RCV1 corpus and store relevant information in Lucene indices for fast feature engineering. The mapping of Reuters documents onto semantic concepts is achieved by generating and applying a classification model exhibiting a multi-class, multi-label functionality.

We deliberately do not apply any threshold, so potentially a document could be assigned to all classes with varying degrees of confidences. The returned class/confidence vector is the new, low-dimensional representation of the document. We decided on Mallet's[3] Naive Bayes implementation to construct classification models for various feature representations including token n-grams. The testing phase handles new data items, i.e. generate feature types, on the fly for calculating similarity values. Before applying the similarity calculation, the new documents need to be mapped onto the semantic concepts. So each document undergoes a process of feature engineering first and is then classified by the trained Naive Bayes model which corresponds to the mapping onto the semantic concepts. Similarity values are then calculated by applying a standard similarity metric, e.g. cosine similarity, to document pairs represented by their affinity to semantic concepts.

[1] http://lucene.apache.org/

[2] http://hunch.net/ vw/

[3] http://mallet.cs.umass.edu/

# 4 Experiments

To evaluate our two approaches to integrate user preferences into the similarity computation, we perform a text classification task using the Reuters RCV1 corpus, a well-known benchmark dataset. The RCV1 dataset ([Lewis *et al.*, 2004]) was drawn from one of the news agency Reuters online databases. The dataset consists of English language stories produced by Reuters journalists between August 20, 1996, and August 19, 1997. To simulate user preferences, we use the stories' topic codes assigned to capture their major subjects. They were organized in four hierarchical groups: CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Government/Social), and MCAT (Markets). Each group is further divided into subgroups providing a more detailed categorization. For the classification task, only documents assigned to exactly one group are considered thereby avoiding a multi-label setting. The annotation process was conducted in a thorough manner - Reuters employed 90 people to handle the annotation of 5.5 million stories per year. We therefore considered the Reuters RCV1 dataset to be an adequate candidate to simulate user preferences.

In both approaches, we experimented with different feature combinations to represent the Reuters documents including unigrams, bigrams, part-of-speech information and named entity information. Sanitization steps included (i) a removal of invalid English words, e.g. a combination of literals and digits, (ii) a removal of stop words and (iii) token stemming using the Porter stemmer [Porter, 1997]. For sentence delimiting and named entity recognition we used Ling Pipe[4] and Apache's OpenNLP[5] natural language processing toolkit. We applied the Stanford part-of-speech tagger to obtain part-of-speech information.

## 4.1 Results

### Similar Document Pairs

To learn a weight vector optimized to separate two classes, we used about 10000 Reuters documents for each of the four main categories, i.e. CCAT, GCAT, ECAT, MCAT. Representative documents were stratified for the positive and negative class. 80% of the documents were used for training, 20% for testing. We point out that by merging two documents with each other, we generate a new example and thus transform the instance space as well. Two documents from the same class are merged into a positive example reflecting a user's decision that these two documents are similar. We handed them over to Vowpal Wabbit's internal linear regression framework. We experimented with different feature representations to learn the weight vector including unigrams, bigrams, nouns, verbs, named entities and combinations thereof. Using the regression framework's performance criteria we compared different feature representations and eventually decided to use only unigrams. To evaluate the discrimination quality of the learnt weight vector on our overall multi-class problem, we used Weka's[6] machine learning framework to compare two settings: once with the learnt weights and once without them. Due to Weka's memory consumption, we used 850 Reuters documents in our classification setting.

Table 1 contrasts the accuracy results (10-fold cross validation) for two classification models, i.e. a Nearest-Neighbor classifier and a linear Support Vector Machine.

| | k = 1 | k = 5 | k = 10 | SVM (lin) |
|---|---|---|---|---|
| Accuracy (unweighted) | 0.64 | 0.58 | 0.60 | 0.89 |
| Accuracy (weighted) | 0.65 | 0.59 | 0.58 | 0.86 |

Table 1: Accuracy results for the Nearest Neighbor classifier and the linear Support Vector Machine(SVM) - once with and once without applying the learnt weights. (10-fold cross validation)

We chose the Nearest Neighbor classification model because it does not apply any additional optimization steps as the Support Vector Machine does. The resulting values state that the learnt weights do not add any additional information regarding the classification problem. We hypothesize that the merging procedure itself strenghtens or weakens the respective dimensions that further weighting is not necessary.

As a second observation we learn that additional processing, e.g. optimization in case of the Support Vector Machine, does allow an increase in classification accuracy. From a theoretical perspective it would be interesting to compare the optimization strategies of (i) using Vowpal Wabbit to learn a weight vector and (ii) using a linear Support Vector Machine to learn Lagrange coefficients - to a certain extent both strategies aim to identify discriminant dimensions in the input space and yet the latter is by far more successful.

### Personal Concept Hierarchy

The second approach's idea is to transform the documents' input space into a space of semantic concepts, i.e. creating a semantic concept representation. To map the documents onto concepts, we first generated a classification model exhibiting a multi-class, multi-label functionality. We decided on Mallet's Naive Bayes implementation to train models for various feature representations including unigrams, bigrams, nouns, verbs and named entities. We used 20000 Reuters documents for each of the four primary-level categories, i.e. CCAT, GCAT, ECAT, MCAT.

Using three of the learnt models, i.e. unigrams, bigrams and named entities, we mapped the Reuters documents onto semantic concepts and performed the multi-class problem with the new semantic concept representation using WEKA's Nearest Neighbour implementation. We evaluated 9000 documents by a 10 fold cross-evaluation - evaluation results for different numbers of neighbours are shown in Table 2.

| | Unigrams | Bigrams | NEs |
|---|---|---|---|
| k = 1 | 0.93 | 0.95 | 0.82 |
| k = 5 | 0.95 | 0.96 | 0.85 |
| k = 10 | 0.95 | 0.97 | 0.86 |

Table 2: Accuracy results for the 4-class classification task based on different number of neighbors and different feature types.

These results show that the semantic concept representation preserves the information and performs well in the simple 4-class classification setting. To create a more realistic setting, we extended the number of concepts by focusing on Reuters secondary level categories. As with the four primary level categories, we used Mallet's classification framework to generate a model for 54 Reuters categories.

[4] http://alias-i.com/lingpipe/

[5] http://opennlp.apache.org/

[6] http://www.cs.waikato.ac.nz/ml/weka/

Since some categories contained only few documents, we decided to use only 100 documents per category as training samples. We used WEKA's Nearest Neighbour classifier implementation to perform the classification task. We evaluated 7500 documents by a 10 fold cross-evaluation - evaluation results are shown in Table 3.

| | Unigrams | Bigrams | NEs |
|---|---|---|---|
| k = 1 | 0.83 | 0.74 | 0.51 |
| k = 5 | 0.83 | 0.74 | 0.47 |
| k = 10 | 0.82 | 0.73 | 0.48 |

Table 3: Accuracy results for the 54-class classification task based on different number of neighbors and different feature types.

In the following, we compared the Nearest Neighbour classifier with two other standard classification schemes - a Naive Bayes classifier and a linear Support Vector Machine.

| | k = 1 | NB | SVM(lin) |
|---|---|---|---|
| Unigrams | 0.83 | 0.74 | 0.83 |

Table 4: Accuracy values for a Nearest Neighbor classifier (k = 1), a Naive Bayes(NB) classifier and a linear Support Vector Machine(SVM).

Table 4 shows similar performance values for the Nearest Neighbor classifier and the Support Vector Machine which indicates that additional optimization does not yield further gains for the classification task.

## 5 Conclusion

In this work we explore two approaches to intuitively integrate user preferences into the similarity computation of textual documents and provide implementation details. Both approaches directly affect the distance metric which has the advantage of being to a certain extent algorithm-independent. Instead of being bound to a certain research field, our approaches can be adopted by algorithms across such fields including Machine Learning or Information Retrieval. The results encourage further engagement and analysis of the underlying ideas. A first direction is to investigate why the learnt optimization weights in the "Similar Document Pairs" approach have so little effect on the resulting accuracy values. From a theoretical perspective a comparison to the optimization strategies of a Support Vector Machine would be interesting. An advantage of the SDP approach certainly is that adding additional document/personal classes is simple. In contrast, the "Personal Concept Hierarchy" approach cannot handle the adding of classes so easily. It has to re-compute the classification models for the mapping operation. As to the requirement of a concept hierarchy for the approach to work, we remark that this information can to a certain degree be automatically generated by taking into account a person's tagging, searching or reading behavior. A natural next step represents the application of both approaches in a real-world setting having persons (i) providing personal information, e.g. in form of decisions, and (ii) evaluating the results and giving feedback.

## Acknowledgments

The work presented in this paper was developed within the DIVINE project (www.weblyzard.com/divine), funded by the Austrian Ministry of Transport, Innovation & Technology (BMVIT) and the Austrian Research Promotion Agency (FFG) within the strategic objective FIT-IT (www.ffg.at/fit-it). The Know-Center is funded within the Austrian COMET Program (Competence Centers for Excellent Technologies) under the auspices of BMVIT, the Austrian Ministry of Economics and Labor, and by the State of Styria.

## References

[Basili *et al.*, 2005] R. Basili, M. Cammisa, and R. Moschitti. A semantic kernel to classify texts with very few training examples. In *Proceedings of the Workshop on Learning in Web Search, at the International Conference on Machine Learning*, 2005.

[Cristianini *et al.*, 2002] N. Cristianini, J. Shawe-Taylor, and H. Lodhi. Latent semantic kernels. *Journal of Intelligent Information Systems*, 18(2-3):127–152, 2002.

[Gabrilovich and Markovitch, 2007] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the International Joint Conference on Artifical intelligence*, pages 1606–1611, 2007.

[Lewis *et al.*, 2004] D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[Metzler and Zaragoza, 2009] D. Metzler and H. Zaragoza. Semi-parametric and non-parametric term weighting for information retrieval. In *Proceedings of the 2nd International Conference on Theory of Information Retrieval*, ICTIR'09, pages 42–53, 2009.

[Porter, 1997] M. Porter. An algorithm for suffix stripping. In *Readings in Information Retrieval*, 1997.

[Qiu and Cho, 2006] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *Proceedings of the International Conference on World Wide Web*, WWW'06, pages 727–736, 2006.

[Radlinski and Joachims, 2007] F. Radlinski and T. Joachims. Active exploration for learning rankings from clickthrough data. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 570–579, 2007.

[Saaty, 2008] T. Saaty. Relative measurement and its generalization in decision making. *RACSAM - Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales.*, 102(2):251–318, 2008.

[Salton and McGill, 1986] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[Shalev-Shwartz *et al.*, 2004] S. Shalev-Shwartz, Y. Singer, and A. Ng. Online and batch learning of pseudo-metrics. In *Proceedings of the International Conference on Machine learning*, ICML'04, pages 94–, New York, NY, USA, 2004.

[Thurstone, 1927] L. Thurstone. A law of comparative judgment. *Psychological Review*, 34:273–286, 1927.

[Weinberger and Saul, 2009] K. Weinberger and L. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009.