

Semantic Similarity in Content-based Filtering

Gabriela Polčicová and Pavol Návrat

Slovak University of Technology
Department of Computer Science and Engineering
polcicova@dcs.elf.stuba.sk, navrat@elf.stuba.sk

Abstract. In content-based filtering systems, content of items is used to recommend new items to the users. It is usually represented by words in natural language where meanings of words are often ambiguous. We studied clustering of words based on their semantic similarity. Then we used word clusters to represent items for recommending new items by content-based filtering. In the paper we present our empirical results.

1 Introduction

Information filtering recommender systems help users to gain orientation in information overload by determining which items are relevant for their interests. One type of information filtering is Content-based filtering (CBF).

In CBF, items contain words in natural language. Meanings of words in natural language are often ambiguous. The problem of word meaning disambiguation is often decomposed to determining semantic similarity of words.

We studied semantic similarity of words and how it can be used in CBF. First, we clustered words from textual items description based on their semantic similarity. Then we used word clusters to represent items. Finally we used those items representations in CBF. To cluster words we used semantic network of English words WordNet¹ [5]. For CBF we used EachMovie and IMDb data.

The rest of the paper is organized as follows. Section 2 and 3 describe content-based filtering and semantic similarity in more detail. Section 4 deals with our approach in using semantic similarity of words in CBF, section 5 describes our experiments and results. Section 6 contains conclusions.

2 Content-based Filtering

To recommend new items for users, CBF follows these steps:

1. *Items representation.* Each item consists of words. First of all, words without meaning (e.g. them, and) - *stop-words* are excluded. Remaining words are stemmed (cut off suffixes). Let us consider a vector representation with dictionary vector \mathbf{D} , where each element d_t is a term (word). Then each document j is represented with a feature vector \mathbf{W}_j , where element w_{jt} is the

¹ We used WordNet version 1.6

weight of word d_t in document j . We use *term frequency-inverse document frequency (tf-idf)*: $w_{jt} = tf(t, j) \log(\frac{n\text{docs}}{df(t)})$, where $tf(t, j)$ is *term frequency* - the number of occurrences of term t in document j , $n\text{docs}$ is the number of all documents and $df(t)$ is *document frequency* - the number of documents containing term t .

2. *User profile creation*. Users assign ratings to the items based of how much they like those items. Profiles of users' interests are generated from items representations and users ratings. They have the same representation as a document, weights are defined by: $\text{profile}_t = \text{profile}_t + \sum_{j=1}^m r'_j w_{jt}$, for each term t . Index j goes through the rated documents, w_{jt} is a weight of the term t in the document j and $r'_j = r_j - \bar{s}$, where \bar{s} is the average of a scale.
3. *Ratings for unrated items estimation*. In order to measure how much a new item matches the profile, we use cosine measure. Its value ranges from -1 to 1 . Let s_n is a number of values in the scale. We divide this interval $\langle -1, 1 \rangle$ into s_n subintervals and we assign each subinterval to one value of a scale. To each item there is assigned an estimated rating e_j according to the subinterval, to which the weight of the value belongs.
4. *Making recommendations*. Items j with $e_j \geq T$ are recommended to the user for a given threshold T .

3 Semantic Similarity of Words

To study relatedness of words, the most often used thesaurus is a semantic network of English words called WordNet [5]. It is lexical reference system organizing words² into synonym sets (synsets), each representing one lexical concept. These synsets are linked by different relations.

There are several types of semantic relatedness. Hierarchical taxonomy expressing IS-A (hypernymy/hyponymy) relation (Figure 1) is considered to be the most appropriate for determining the semantic similarity [7], [3]. It can be used by two main approaches: edge-based and node-based [3].

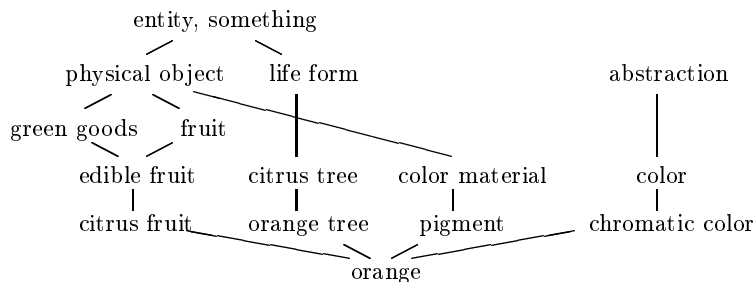


Fig. 1. Simplified hypernym hierarchy for word "orange".

² A lexical category of the word must be determined.

3.1 Edge-based Approach

Edge-based approach measures minimal distance³ between concepts (synsets) in a hierarchical structure. Resnik [7] presents edge-based measure that converts the minimal length between concepts c_1 and c_2 (c_i is the concept (synset) that represents one sense of a word w_i , $i = 1, 2$). It is given by:

$$\text{sim}_R^e(w_1, w_2) = 2 \times \text{MAX} - \left[\min_{c_1, c_2} \text{len}(c_1, c_2) \right], \quad (1)$$

where MAX is maximum depth of the taxonomy and $\text{len}(c_1, c_2)$ is length of the shortest path between concepts c_1 and c_2 [7], [3].

3.2 Node-based Approach

In addition to hierarchical taxonomy, node-based approach uses a large text corpus to compute probabilities $p(c)$ of encountering an instance of concept c and then its information content $-\log(p(c))$. The idea behind this approach is that similarity between concepts should be proportional to "the extent to which they share information". There are several similarity measures presented in the literature [7], [3], [4]. One of them is Lin's similarity measure [4]

$$\text{sim}_L^n(w_1, w_2) = \frac{2 \log(p(\text{lso}(c_1, c_2)))}{\log(p(c_1)) + \log(p(c_2))}, \quad (2)$$

where $\text{lso}(c_1, c_2)$ is the lowest super-ordinate of word concepts c_1 and c_2 .

4 Using Semantic Similarity of Words in Content-based Filtering

To use semantic similarity of words in CBF, we followed these steps:

1. *Preprocessing*. Textual documents (items) were modified so that each sentence was just in one line. This was needed for the next step.
2. *Lexical categories assignment*. To do this, a part-of-speech tagger was used.
3. *Selecting nouns and verbs*. Nouns and verbs were selected to create a list of all nouns and a list of all verbs. It was done because we assumed that just they retain the main meaning of the sentence, similarly to [9], [7], [1].
4. *Lists of synsets assignment*. To each selected word we assigned a list of hypernym synsets from WordNet IS-A taxonomy. In the list, we included only hypernym synsets that are on a path to its root synset with length greater or equal to threshold L . This was done to avoid too "general" relationships.
5. *Synset frequencies computation*. Synset frequencies were computed from lists of hypernym synsets. This step was needed only for node-similarities.

³ Note, that word's senses can belong to more than one concept and that there can be more than one path that links two concepts.

6. *Semantic similarities computation.* To compute similarities among nouns and among verbs from lists of synsets, we used $\text{sim}_R^e(w_1, w_2)$ (1) and $\text{sim}_L^n(w_1, w_2)$ (2) measures. To compute synset probabilities, synset frequencies from step 5 were used. Lists of synsets was used to create lists of different nouns and verbs in order to avoid using several forms of one word (e.g. boy, boys).
7. *Converting similarities to dissimilarities.* Similarities were transformed to dissimilarities by using $\text{dis}(w_1, w_2) = 1.0 - (\text{sim}(w_1, w_2)/\text{max})$, where max is the maximal possible similarity. For edge-based similarity, $\text{max} = 2 \times$ total depth of hypernym network, max value for node-based similarity is 1.0.
8. *Nouns and verbs clustering.* Hierarchical clustering with complete agglomerative method was used to cluster nouns and then verbs. We used several values for N (number of noun clusters) and V (number of verb clusters).
9. *Creating semantic representation.* Each noun and verb was replaced by its cluster. Since proper names cannot be clustered based on semantics, in addition to those clusters we used proper names. To their number we refer as to P . Thus, noun clusters $Cn_i (i = 1, \dots, N)$, verb clusters $Cv_i (i = 1, \dots, V)$ and proper names $Pn_i (i = 1, \dots, P)$ created a dictionary vector $\mathbf{D} = (Cn_1, \dots, Cn_N, Cv_1, \dots, Cv_V, Pn_1, \dots, Pn_P)$. Vectors of items contained numbers of occurrences of words for each cluster and numbers of occurrences of proper names. To this representation we further refer as to *semantic representation*.
10. *Running CBF.* We run content-based filtering for pure *tf-idf* items representation and *semantic representations* of items (steps 2-4 from section 2).

5 Experiments

5.1 Data

In our experiments we used 2 datasets. The first one is EachMovie database⁴. It contains explicit ratings for movies (2811983 ratings from 72916 users to 1628 movies). We transformed rating scale to integers $1, \dots, 6$. The second dataset is Internet Movie Database⁵ (IMDb) containing movie descriptions.

5.2 Parameter Settings

To create *tf-idf* representation of a movie, the first 3 actors, first director, title and textual description were selected from IMDb descriptions. Then we followed step 1, section 2. Porter algorithm was used to stem words [6]. Number of words (elements of \mathbf{D}) was 16467. Since we used rating scale $1, \dots, 6$, T was set to 4.

To create *semantic* representation, we used IMDb description as for *tf-idf* representation, but we excluded titles. We did so, because tagger could be applied only to the whole sentences, what titles usually are not. We followed steps 1-9 from section 4. Brill's tagger [2] was applied in step 2. L was set to 3 (step 4) and

⁴ <http://www.research.digital.com/SRC/eachmovie/>

⁵ <http://www.imdb.com/>

thus $N = 4594$ and $V = 1696$ (step 6). In edge-similarity, the maximal length of path to the root synset was 20 ($max = 40$) (step 7). The number of proper names $P = 3942$ (step 9).

Table 1. Labels of datasets used for CBF.

	$N = 500$	$N = 700$	$N = 2000$	$N = 2500$
Representation	$V = 500$	$V = 700$	$V = 1000$	$V = 1000$
edge-based semantic	A	C	E	G
node-based semantic	B	D	F	H
tf-idf	T			

We experimented with data in order to select several meaningful values for N and V . Our task was not to find the appropriate number of clusters, but to study whether verbs and nouns clustering is helpful in CBF. We present results achieved on 9 datasets for 4 different N and V values (Table 1).

5.3 Results and Discussion

10-fold cross-validation was used to evaluate the quality of estimations. In each step of cross-validation 10% of each user’s ratings were assigned to *test* set and remaining 90% to *training set*. CBF with each representation run with using the same test and training sets. To evaluate the results we used *Mean absolute error (MAE)* and *F-measure* [8].

Results are presented in figure 2. They indicate that CBF with *tf-idf* and with *semantic* representation provide comparable results. We applied ANOVA with the Bonferroni procedure on 95% level to evaluate the results. For *F-measure* test showed no significant difference. Evaluated with *MAE*, CBF with datasets A and B achieved significantly better results than CBF with E, F, G, H, T and CBF with D significantly outperform CBF with H and T datasets.

The results indicate that for appropriate number of clusters CBF with *semantic* representation might outperform CBF with *tf-idf* representation. However we should like to see a stronger evidence to this hypothesis. To discuss reasons for the results, let us repeat several simplifications, we made: (1) we could not use titles for *semantic* representation but we use them in *tf-idf* representation. (2) we did not use any algorithm to choose the appropriate number of noun and verb clusters. (3) we could not evaluate and correct errors of part-of-speech tagger. We consider these simplifications to be important and we assume that they affect the results.

6 Conclusions and the Future Work

We used semantic similarity to cluster verbs and nouns from textual items to create *semantic* representation for those items. We compared results of CBF with

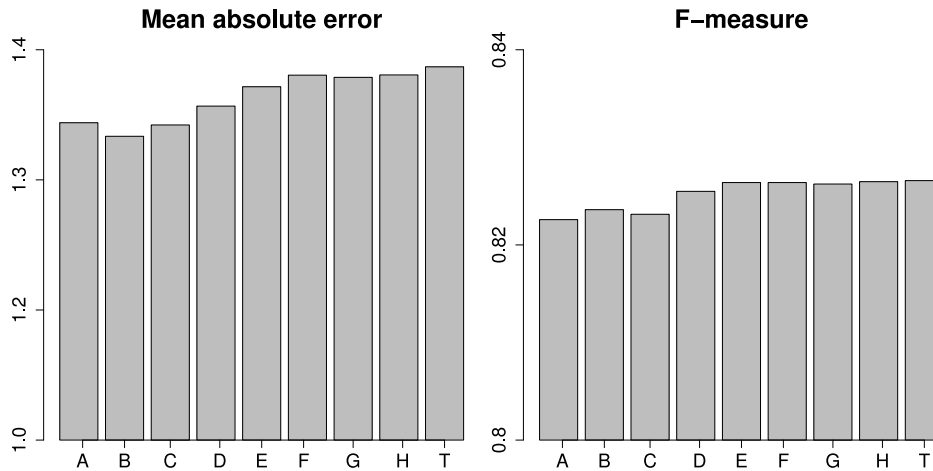


Fig. 2. CBF results for with *semantic* representation (datasets A, . . . , H) and with *tf-idf* representation (dataset T), evaluated using *MAE* and *F-measure*.

commonly used *tf-idf* and with our *semantic* representation. Evaluated with *F-measure*, CBF with *semantic* representation provided no significant difference from CBF with *tf-idf* representation. For certain numbers of clusters, CBF with *semantic* representation provided significantly smaller *mean absolute error*.

In the future we plan to study methods for determining the appropriate number of noun and verb clusters and experiment with applying part-of-speech taggers to determine lexical categories to words in titles.

References

1. A. Arampatzis, P. Th. C. van der Weide, and P. Koster. Text filtering using linguistically-motivated indexing terms, 1999.
2. E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, 1992.
3. J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Int'l Conf. on Research on Computational Linguistics, Taiwan, 1997*.
4. D. Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, 1998.
5. G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K.J. Miller. Introduction to wordnet: An on-line lexical database. *Journal of Lexicography*, 3(4):234–244, 1990.
6. M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
7. P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453, 1995.
8. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender systems—a case study, 2000.
9. S. Scott and S. Matwin. Text classification using WordNet hypernyms. In S. Harabagiu, editor, *Use of WordNet in Natural Language Processing Systems*, pages 38–44. Association for Computational Linguistics, 1998.