Near-Duplicate Detection for Online-Shops Owners: an FCA-based Approach

Dmitry I. Ignatov, Andrey V. Konstantiov, and Yana Chubis

National Research University Higher School of Economics, Moscow, Russia
dignatov@hse.ru
http://www.hse.ru

Abstract. We proposed a prototype of near-duplicate detection system for web-shop owners. It's a typical situation for this online businesses to buy description of their goods from so-called copyrighters. Copyrighter can cheat from time to time and provide the owner with some almost identical descriptions for different items. In this paper we demonstrated how can we use FCA for fast clustering and revealing such duplicates in real online perfume shop's datasets.

Keywords: Near duplicate detection, Formal Concept Analysis, E-commerce

1 Introduction

Finding near-duplicate documents on the Internet is a well-studied problem, which necessitates creation of efficient methods for computing clusters of duplicates [1, 2, 5, 6, 8]. The origin of duplicates can be different: from intended duplicating information on several severs by companies (legal mirrors) to cheating indexing programs of websites, illegal copying and almost identical spammer messages. However, the aim of the paper is to provide an average web-shop owner with an effective means of near-duplicate detection in the description of the shop items. These duplicates appear because of unfair copyrighters who provide the web-shop owner similar content's descriptions.

Usually duplicates are defined in terms of similarity relation on pairs of documents: two documents are similar if a numerical measure of their similarity exceeds a certain threshold (e.g., see [1]). The situation is represented then by a graph where vertices are documents and edges correspond to pairs of the similarity relation. Clusters of similar documents are computed then as cliques or as connected components of such similarity graphs [1].

In this paper we consider similarity not as a relation on the set of documents, but as an operation taking each two documents to the set of all common elements of their concise descriptions. Here description elements are syntactical units (*shingles*). To this end we employed an approach based on formal concepts: Clusters of documents are given by formal concepts of the context where objects correspond to description units (units of a language describing documents, e.g. shingles) and attributes are document names. A cluster of *very similar documents* corresponds then to a formal concept such that the size of extent (the number of common description units of documents) exceeds a threshold given by parameter. Thus, generating very similar documents is reduced to the problem of Data Mining [9] known as generating *frequent closed itemsets*. In this paper we compare results of its application (for various values of thresholds) with the list of duplicates obtained by applying other methods to the same collection of documents.

2 Main part

For creating document images we used standard syntactical approach with different parameters, detailed description of which can be found in [1]. For each text the program **shingle** with two parameters (*length* and *offset*) generate contiguous subsequences of size *length* such that the distance between the beginnings of two subsequent substrings is *offset*. The set of sequences obtained in this way is hashed so that each sequence receives its own hash code. From the set of hash codes that corresponds to the document a fixed size (given by parameter) subset is chosen by means of random permutations described in [1]. The probability of the fact that minimal elements in permutations on hash code sets of shingles of documents A and B (these sets are denoted by F_A and F_B, respectively) coincide, equals to the similarity measure of these documents $sim(A, B)$: $sim(A, B) = P[min\{\pi(F_A)\} = min\{\pi(F_B)\}] = \frac{|F_A \cap F_B|}{|F_A \cup F_B|}$.

First, we briefly recall the main definitions of Formal Concept Analysis (FCA) [3]. Let G and M be sets, called the set of objects and the set of attributes, respectively. Let I be a relation $I \subseteq G \times M$ between objects and attributes: for $g \in G$, $m \in M$, gIm holds iff the object g has the attribute m. The triple $K = (G, M, I)$ is called a *(formal) context*. Formal contexts are naturally given by cross tables, where a cross for a pair (g, m) means that this pair belongs to the relation I. If $A \subseteq G$, $B \subseteq M$ are arbitrary subsets, then *derivation operators* are given as follows:

$$A' := \{m \in M \mid gIm \text{ for all } g \in A\},$$
$$B' := \{g \in G \mid gIm \text{ for all } m \in B\}.$$

The pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$ is called a *(formal) concept (of the context K)* with *extent* A and *intent* B.

The operation $(\cdot)''$ is a closure operator, i.e., it is idempotent ($X'''' = X''$), extensive ($X \subseteq X''$), and monotone ($X \subseteq Y \Rightarrow X'' \subseteq Y''$). Sets $A \subseteq G$, $B \subseteq M$ are called *closed* if $A'' = A$ and $B'' = B$. Obviously, extents and intents are closed sets. Formal concepts of context are ordered as follows: $(A_1, B_1) \leq (A_2, B_2)$ iff $A_1 \subseteq A_2 (\Leftrightarrow B_1 \supseteq B_2)$. With respect to this order the set of all formal concepts of the context K makes a lattice, called a *concept lattice* $\mathfrak{B}(K)$ [3].

A set $B \in M$ is called *k-frequent* if $|B'| \leq k$ (i.e., the set of attributes B occurs in more than k objects), where k is parameter. Computing frequent closed sets of attributes (or itemsets) became important in Data Mining since these sets give the set of all association rules [9]. For our implementation where contexts are given by set G of description units (e.g., shingles), set M of documents and incidence (occurrence) relation I on them, we define a cluster of k-similar documents as intent B of a concept (A, B) where $|A| \geq k$. Although the set of all closed sets of attributes (intents) may be exponential with respect to the number of attributes, in practice contexts are *sparse* (i.e., the average number of attributes per object is fairly small). One of the leaders of Frequent Itemset Mining Implementations (FIMI) in time efficiency was the algorithm FPmax* [4]. We used this algorithm for finding similarities of documents and generating clusters of *very similar documents*. As mentioned before, objects are description units (shingles or words) and attributes are documents. For representation of this type *frequent closed itemsets* are closed sets of documents, for which the number of common description units in document images exceeds a given threshold. Actually, FPmax* generates *maximal frequent itemsets*, i.e., closed frequent itemsets that are maximal by set inclusion.

Software for experiments with syntactical representation comprise the units that perform the following operations: 1) Generating shingles with given parameters length-of-shingle, offset; 2) Hashing shingles; 3) Composition of document image by selecting subsets (of hash codes) of shingles; 4) Composition of the inverted table the list of identifiers of documents shingle thus preparing data to the format of programs for computing closed itemsets; 5) Computation of clusters of *k-similar documents* with FPmax* algorithm: the output consists of strings, where the first elements are names (ids) of documents and the last element is the number of common shingles for these documents; 6) Comparing results with the existing list of duplicates (in our experiments with the SpellSmellExpert collection of documents).

In our experiments we used three text collections: RUS, SpellSmell, SpellSmell-Expert. RUS was composed from 9 original texts of Russian literature and the rest 10 was produced from them by near-duplicate generator. SpellSmell contains 3500 perfume descriptions from the online web-shop spellsmell.ru. SpellSmellExpert contains 70 near-duplicate descriptions from SpellSmell confirmed by Experts. We use the first collection mainly for preliminary testing (Table 1). The image of a document has a length $n = 1000$ in the provided experiments.

To compare results of clustering of our approach with Cluto (one of the best document clustering packages) we chose the repeated-bisecting algorithm that uses the cosine similarity function with a 10-way partitioning (ClusterRB), which is mostly scalable according to its author [7]. The number of clusters is a parameter, documents are given by sets of attributes, fingerprints in our case. The algorithm outputs a set of disjoint clusters.

Even though both algorithms have almost the same elapsed time in our experiments, FPmax showed better results in terms of F1 measure (Table 2).

As a result of this small research the online copyrighter cabinet was developed by SpellSmell.ru for uniqueness checking of uploaded text collections written by copyrighters. According to SpellSmell owners' opinion the implementation is fast, scalable and detected duplicates are relevant.

Table 1. FPmax* clustering results of the RUS collection

Precision	Recall	Threshold	F1
1	0.1	900	0.18
1	0.3	800	0.33
1	0.33	700	0.5
1	0.9	500	0.9
1	1	400	1

Table 2. Comparison of clustering results of FP-max and Cluto on the SpellSmell collection

FPmax*				
Time,s	Precision	Recall	Threshold	F1
0.1	**0.7**	0.07	900	0.12
0.1	0.5	0.08	800	0.13
0.2	0.42	0.1	700	0.16
0.3	0.3	0.2	500	0.12
0.5	0.28	0.3	400	0.28
0.7	0.27	0.4	300	**0.32**
1.9	0.2	**0.6**	200	0.3

Cluto				
Time,s	Precision	Recall	Number of clusters	F1
0.1	0.01	**0.91**	69	0.02
0.2	0.02	0.72	193	0.04
1.5	**0.84**	0.1	1812	**0.18**

References

1. Broder,A.: Identifying and Filtering Near-Duplicate Documents, in R. Giancarlo and D. Sankoff, Proc. Annual Symposium on Combinatorial Pattern Matching, Lecture Notes in Computer Science, vol. 1848, 2000.
2. Chowdhury,A., Frieder, O., Grossman,D.A. and McCabe,M.C.: Collection statistics for fast duplicate document detection, ACM Transactions on Information Systems, 20(2): 171-191, 2002
3. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations, Springer, 1999.
4. Grahne, G., Zhu, J.: Efficiently Using Prefix-trees in Mining Frequent Itemsets, in Proc. FIMI03 Workshop, 2003.
5. Haveliwala,T. H., Gionis,A., Klein,D., Indyk,P.: Evaluating Strategies for Similarity Search on the Web, in Proc. WWW'2002, Honolulu, 2002, pp.432-442.
6. Ilyinsky,S., Kuzmin,M., Melkov,A., Segalovich, I.: An efficient method to detect duplicates of Web documents with the use of inverted index, in Proc. 11th Int. World Wide Web Conference (WWW'2002), ACM, 2002
7. Karypis, G.: CLUTO. A Clustering Toolkit. University of Minnesota, Department of Computer Science Minneapolis, MN 55455, Technical Report: 02-017, November 28, 2003
8. Kolcz, A., Chowdhury,A., Alspector,J.: Improved Robustness of Signature-Based Near-Replica Detection via Lexicon Randomization, in EDs. W.Kim, R. Kohavi, J. Gehrke, W. DuMouchel, Proc. KDD'04, Seattle, 2004, pp.605-610.
9. Pasquier,N., Bastide,Y., Taouil,R., Lakhal,L.: Efficient Mining of Association Rules Using Closed Itemset Lattices, Inform. Syst., 24(1), 25-46, 1999.