

Tagging Artists using Co-Occurrences on the Web

Gijs Geleijnse Jan Korst

Philips Research, High Tech Campus 34, 5656 AE Eindhoven, The Netherlands
gijs.geleijnse@philips.com , jan.korst@philips.com

Abstract

We present an efficient unsupervised approach in finding subjective artist meta-data on the world wide web. Since we are interested in the collective knowledge on artists as available on the web, our method is based on the extraction of information from multiple web pages. We use co-occurrences of pairs of artists on the web to identify similarity between artists. To determine the applicability of tags to artists we follow the same approach. We use Google to find the co-occurrences on the web, either by analyzing Google excerpts found by querying patterns or by scanning full documents. Since the same tags are often applicable to related artists, we use similarity between artists to improve the tagging. We tested and compared the two co-occurrence extraction methods on two different domains: finding the most appropriate genres for music artists, and finding art-styles for painters. The results are convincing and show that the use of similar artists indeed improves the precision of the tagging.

Key words: information extraction, Google, co-occurrence analysis, tagging, artist similarity.

1 Introduction

Popular internet radio stations, such as *last.fm*, provide their users with the possibility to annotate their music. On the one hand users can indicate similarity between artists. For example a user can indicate that *U2* is similar to *Coldplay*. On the other hand, users are offered the opportunity to tag songs, artists and albums (Figure 1). Examples of common tags are *singer-songwriter*, *german* and *sexy*. Using the *collective knowledge* of the users, an artist or album is characterized by the tags mostly assigned to it (O'Reilly, 2005).

These tags and indications of similarity can provide meaningful data when browsing through a collection. Moreover, they help people to discover new music. In addition to audio-extracted features, subjective meta-data such as tags and similarity between artists are useful for a music recommender.

The web is currently the de-facto source of information on popular music. Fan sites, artist homepages, webpages with playlists etc. give valuable information on virtually every artist. In this paper we present a method to compute (a) a measure of similarity between artists and (b) a measure of applicability of tags to artists. We extract such meta-data from the web using Google. Using multiple web pages, we want to compute such subjective meta-data using the collective knowledge on the web. In line with the theory of latent semantic analysis (Manning & Schütze, 1999) we use the paradigm that related terms frequently co-occur in the same context.

This paper is organized as follows. In the next section we present the problem description, followed by a brief overview of related work in Section 3. In Section 4 we discuss two efficient methods to extract co-occurrences from the web. Such co-occurrences are used to compute a measure of similarity between artists in Section 5. Using a list of tags, we use co-occurrences

and organizations. The idea of extracting relations using patterns is similar to one of the methods presented here. However, in Snowball the relations gathered are not evaluated.

Cimiano and Staab (2004) describe a method to use a search engine to verify a hypothesis relation. For example, if we are interested in the ‘is a’ or hyponym relation and we have the instance *Nile*, we can use a search engine to query phrases expressing this relation (e.g. “*rivers such as the Nile*” and “*cities such as the Nile*”). The number of hits to such queries is used to determine the validity of the hypothesis. Per instance, the number of queries is linear in the number of classes (e.g. *city* and *river*) considered.

In (Ravichandran & Hovy, 2002) a technique is introduced to find precise patterns using a training set and a large text corpus. In (Geleijnse & Korst, 2006a) a method is presented to extract information from the web using effective patterns. Search engine queries are constructed by combining a pattern (e.g. *was born in*) with an instance (e.g. *Anton Philips*) into a phrase (e.g. “*Anton Philips was born in*”). Such queries lead to precise search results, from which the related instances (e.g. *Zaltbommel*) can be extracted and reused in different queries (“*was born in Zaltbommel*”).

The number of Google *hits* for pairs of terms can be used to compute a semantic distance between terms (Cilibrasi & Vitanyi, 2004). The nature of the relation is not identified, but the technique can for example be used to cluster painters. In (Zadel & Fujinaga, 2004) a similar method is used to cluster artists using search engine counts. In (Schedl *et al.*, 2005a), the number of Google hits of combinations of artists is used in clustering artists. The same co-occurrence count technique is recently used in (Schedl *et al.*, 2006) to assign genres to artists.

KnowItAll is a hybrid named-entity extraction system (Etzioni *et al.*, 2005) that finds lists of instances of a given class from the web using a search engine. It combines hyponym patterns (Hearst, 1992) and class-specific, learned patterns to identify and extract named entities. Moreover, it uses adaptive wrapper algorithms (Crescenzi & Mecca, 2004) to extract information from HTML markup such as tables. Contrary to our method, it does not use instances to formulate queries. In (Downey *et al.*, 2005) the information extracted by KnowItAll is evaluated using a combinatorial model based on the redundancy of information on the web.

In (Boer *et al.*, 2006) a number of documents on art styles are collected. Names of painters are identified within these documents. The documents are evaluated by counting the number of painters in a training set (of e.g. *expressionists*) that appear in the document. Painters appearing on the best ranked documents are then mapped to the style. De Boer *et al.* use a training set and page evaluation, where we simply observe co-occurrences.

In (Knees *et al.*, 2004) clusters of artists are identified using similar terms on web pages describing the artists. The same *tf-idf* approach is used in (Brooks & Montanez, 2006) to automatically tag weblogs.

4 Efficiently finding co-occurrences on the web

When two terms u and v co-occur relatively often in the same context (e.g. within the same web page), they can be assumed to be related (Manning & Schütze, 1999). With respect to the Web, we expect terms to be related if a search engine retrieves many pages containing both terms. In earlier work search engine queries were used that contained both the terms (e.g. (Cimiano & Staab, 2004; Cilibrasi & Vitanyi, 2004; Schedl *et al.*, 2005b)). The number of Google hits to such queries were used as co-occurrence count $co(u, v)$.

In (Geleijnse & Korst, 2006b) and (Geleijnse *et al.*, 2006) we observed two drawbacks of such an approach. On the one hand this approach leads to a high Google Complexity, i.e. many queries to a search engine are required. If we are interested in co-occurrences of terms in two sets of sizes

n and m , this results in $O(n \cdot m)$ queries. On the other hand, the estimated numbers of hits to queries can be unreliable (Véronis, 2006). Our earlier work shows that an approach using the number of Google hits as co-occurrence measure between two terms indeed does not give satisfactory results, we focus on more efficient methods in this work (Geleijnse & Korst, 2006b; Geleijnse *et al.*, 2006).

In this paper, we aim for an approach in which the number of queries is linear in the number of terms in the sets. In this section, we present two alternative methods to acquire co-occurrence counts using a linear Google Complexity. Earlier work showed that these alternatives yield better results than a straight-forward quadratic approach where the number of co-occurrences of terms a and b equals the total number of hits to a query containing the two terms.

We next discuss a pattern-based approach (Section 4.1) and a document-based approach (Section 4.2) to acquire co-occurrence counts. In Section 5 we use the number of co-occurrences of pairs of artists to compute a similarity score $t(a, b)$ for all artist pairs. The co-occurrence counts of artists and tags are used in Section 6.

4.1 Pattern-based co-occurrence count (PAT)

The co-occurrence count using patterns (PAT) is based on occurrences of terms in phrases on the web (Hearst, 1992; Korst *et al.*, 2006). We observe combinations of terms in phrases that express the relation we are interested in. For example, if we are interested in the relation between music artists and their genres, an appropriate phrase that links terms of the two could be “[*artist*] is one of the biggest [*genre*] artists”.

We can identify these patterns automatically by using a training set T of similar artists and tags (Ravichandran & Hovy, 2002; Geleijnse & Korst, 2006a). Learning patterns can be done with $O(|T|)$ queries.

We can learn patterns by querying combinations of related terms (e.g. *ABBA* and *disco*) and observe which text fragments relate the two terms. However, not all patterns found will provide useful results. For example, the phrase “*ABBA Fridge Magnets ‘Disco’!*” leads to the pattern “[*artist*] *Fridge Magnets* [*tag*]” which may not be suited to find other artist-tag co-occurrences. Therefore, we select the patterns that are likely to provide many useful results, by testing the patterns found on effectiveness. The best ranked patterns, based on both precision and recall, are considered to be the most effective ones.

We also construct a set of patterns expressing similarity between artists in A . This approach gives us the opportunity to specify the relatedness between artists. For example, we can be solely interested in artists who played together. A pattern such as “[*artist*] recorded a duet with [*artist*]” could be suitable for this purpose.

We use combinations of a pattern and an instance or a category as a query to the search engine (Geleijnse & Korst, 2006a). For example, if we have the pattern “[*tag*] artist such as [*artist*]”, we use “artist such as” in queries in combinations with all tags and artists. We use this pattern e.g. both for the query “Country artists such as” and for the query “artists such as Prince”. In the excerpts found with the first query, we identify artists in A related to *country*, while in the results for the second query we search for tags in L related to *Prince*.

These queries provide access to relevant data. From the excerpts returned by the search engine, we thus identify the elements of either A or L to measure the number of co-occurrences of the pairs. Hence, for terms u and v the co-occurrence $\text{co}(u, v)$ is defined as follows using PAT.

$$\text{co}(u, v) = \text{‘number of occurrences of } u \text{ by querying patterns with } v \text{’} + \\ \text{‘number of occurrences of } v \text{ by querying patterns with } u \text{’}$$

Retrieving co-occurrences of tags and artists requires $m \cdot (|A| + |L|)$ queries, when using m patterns. We perform $k \cdot |A|$ queries to generate the data for finding the co-occurrences for relating artists in A , where k is the number of patterns expressing relatedness between two artists in A . Considering m and k as constants, the total amount of queries for this method is thus $O(|A| + |L|)$.

4.2 Document-based co-occurrence count (DOC)

An alternative technique to efficiently find co-occurrence counts on the web focusses on observing co-occurrences within documents (DOC). Here, we collect the first k URLs of the documents returned by the search engine for a given query. These k URLs are the most relevant for the query submitted based on the ranking used by the search engine (Brin & Page, 1998).

For each term in the sets A and L we collect the top k documents. For artists in A , we retrieve each document using the URLs found by the search engine. We count the occurrences of the tags in L in the retrieved documents. From the documents retrieved by querying a tag in L , we similarly extract the occurrences of artists in A .

The documents obtained using DOC are the most relevant for each element $b \in A$. For the artists queried we expect biographies, fan pages, pages of museums, entries in database sites and so on. The tags in L (e.g. the genres or styles) mentioned in these pages will most probably reflect the genre of the artist queried.

The co-occurrences function is here thus defined as follows.

$$\text{co}(u, v) = \text{‘number of occurrences of } u \text{ in documents found when querying } v\text{’} + \text{‘number of occurrences of } v \text{ in documents found when querying } u\text{’}$$

To find co-occurrences of artists and tags $O(|A| + |L|)$ queries are required. The documents downloaded with the queries consisting of artist names can be reused when retrieving co-occurrences of pairs of artists. Consequently, this method thus requires only $O(|A| + |L|)$ queries. However, additional data communication is required since for each query up to k documents have to be downloaded instead of using only the data provided by the search engine.

5 Computing artist similarities

Using either PAT or DOC, we acquire co-occurrence counts for pairs $(a, b) \in A \times A$.

Per artist $a \in A$ we could consider the artist $b \in A$ with the highest co-occurrence count $\text{co}(a, b)$ to be the most similar to a . However, we observe that, in that case, frequently occurring artists in A have a relatively large probability to be related to any artist. This observation leads to a normalized approach, inspired by the theory of pointwise mutual information (Manning & Schütze, 1999; Downey *et al.*, 2005).

Per pair $(a, b) \in A \times A$, with $\text{co}(a, b) \geq 1$, we compute the scores $T(a, b)$ and $T(b, a)$.

$$T(a, b) = \frac{\text{co}(a, b)}{\sum_{c, c \neq b} \text{co}(c, b)} \quad (1)$$

Note that, unlike $\text{co}(a, b)$, $T(a, b)$ is not symmetric in its arguments. The co-occurrence count $\text{co}(a, a)$ is undefined for all $a \in A$, since no meaningful values can be identified using any of the co-occurrence extraction methods. As a result, $T(a, a)$ is also undefined for all a .

We introduce a normalized score t such that $t(a, b)$ can be read as the probability that artist b is similar to a .

$$t(a, b) = \frac{T(a, b)}{\sum_{c, c \neq a} T(a, c)} \quad (2)$$

If we are interested in the artist most similar to a , we select the b for which $t(a, b)$ is maximal.

6 Tagging artists

Using the co-occurrence counts $\text{co}(a, l)$ of artists $a \in A$ and tags $l \in L$, we can find similar scoring functions $S(a, l)$ for all pairs (a, l) such that $\text{co}(a, l) \geq 1$.

$$S(a, l) = \frac{\text{co}(a, l)}{\sum_b \text{co}(b, l)} \quad (3)$$

Again, we normalize this function.

$$s(a, l) = \frac{S(a, l)}{\sum_h S(a, h)} \quad (4)$$

Now, as a first approach, we could use $s(a, l)$ to compute the n most appropriate tags per artist. However, since we also compute the similarities between artists, we can incorporate this information to improve the tagging. We here use the hypothesis that for an artist a tags applicable to closely related artists are also applicable to a .

In earlier work, (Geleijnse & Korst, 2006b; Geleijnse *et al.*, 2006), we assumed the relation between artists and tags to be functional. That is, each artist was assumed to be related to at most one tag. We identified a list of the best applicable tags for artist a and its k nearest neighbors. For a final mapping between a and L , we selected the most occurring tag in the list.

In this paper however, we assume that multiple tags can be applicable to one artist. We are therefore interested in the probability that a tag l is applicable to artist a . The majority voting technique as used in earlier work is thus not sufficient, we however do want to take similar artists into account.

The degree of similarity of some artist b to a was given by $t(a, b)$. For tag l , the relatedness between b and l is given by $s(b, l)$. If b is closely related to a , we want $s(b, l)$ to contribute significantly to the final score $p(a, l)$. Using the normalized scoring functions, we can compute the relatedness of tag l to artist a as follows.

$$p'(a, l) = \sum_{b, b \neq a} t(a, b) \cdot s(b, l) \quad (5)$$

If for $s(a, l)$ erroneously a high score is found, this error is decreased when the close related artists b have low scores for $s(b, l)$.

Since $t(a, a)$ is not defined and we consider $s(a, l)$ relevant when computing the scores for the tags with respect to artist a , we introduce a weight w for $s(a, l)$ as a substitute for $t(a, a)$.

$$p(a, l) = w \cdot s(a, l) + (1 - w) \cdot \sum_{b, b \neq a} t(a, b) \cdot s(b, l) \quad (6)$$

Note that $p(a, l) = s(a, l)$ for $w = 1$. For artist a and tag l , $p(a, l)$ can be read as the probability that tag l is applicable to a . Now $p(a, l)$ is a normalized scoring function for the applicability of tag l to artist a .

$$\begin{aligned}
\sum_l p(a,l) &= \sum_l (w \cdot s(a,l) + (1-w) \cdot \sum_{b,b \neq a} t(a,b) \cdot s(b,l)) \\
&= \sum_l (w \cdot s(a,l)) + \sum_l ((1-w) \cdot \sum_{b,b \neq a} t(a,b) \cdot s(b,l)) \\
&= w \cdot \sum_l s(a,l) + (1-w) \cdot \sum_l \sum_{b,b \neq a} t(a,b) \cdot s(b,l) \\
&= w + (1-w) \cdot \sum_l \sum_{b,b \neq a} t(a,b) \cdot s(b,l) \\
&= w + (1-w) \cdot \sum_{b,b \neq a} \sum_l t(a,b) \cdot s(b,l) \\
&= w + (1-w) \cdot \sum_{b,b \neq a} t(a,b) \cdot \sum_l s(b,l) \\
&= w + (1-w) \cdot \sum_{b,b \neq a} t(a,b) \\
&= w + (1-w) \\
&= 1
\end{aligned}$$

It now remains to find an appropriate value for w . One approach is to identify a training set of artists and related tags. Using the co-occurrences acquired we can determine the value of w , $0 \leq w \leq 1$, for which the scores of the tags best fit the training set. In the following section, we investigate whether the performance of the artist tagging method indeed improves for values of w smaller than 1.

7 Experimental results

We present two experiments. The first experiment is in the music domain where we find artist similarity among list of 224 artists. The artists are subdivided into 14 genres. We test whether the most applicable tag corresponds to the genre in the reference list (Knees *et al.*, 2004). In the second experiment, we construct a list of painters and a list of art-styles using Wikipedia and link the two.

In these experiments, we assume that per artist one tag is best applicable. We therefore only test the performance of the best scoring tag. In future work however, we plan to test our approach by creating a test set where multiple tags are applicable to an artist.

7.1 Tagging artists with genres

In the first experiment, A is the set of all artists and L the set of all genres in the list composed by Knees *et al.* (Knees *et al.*, 2004). This list consists of 14 genres, each with 16 artists. The genres mentioned in the list are not all suitable for finding co-occurrences. For example, the term *classical* is ambiguous and *Alternative Rock/Indie* is an infrequent term. The set of tags L therefore consists of manually rewritten names of the genres (such as *classical music* instead of *classical*). Moreover, some synonyms were added. After collecting the numbers of co-occurrences of artists and genres, we summed up the scores of the co-occurrences for synonyms. Thus, for each artist a the number of co-occurrences with the terms *Indie* and *Alternative Rock* are added to the co-occurrences of a with the genre *Alternative Rock/Indie*.

We performed the experiments using both PAT and DOC to obtain the co-occurrences. For PAT, we selected 16 learned patterns for finding co-occurrences of the elements in A and L . For learning,

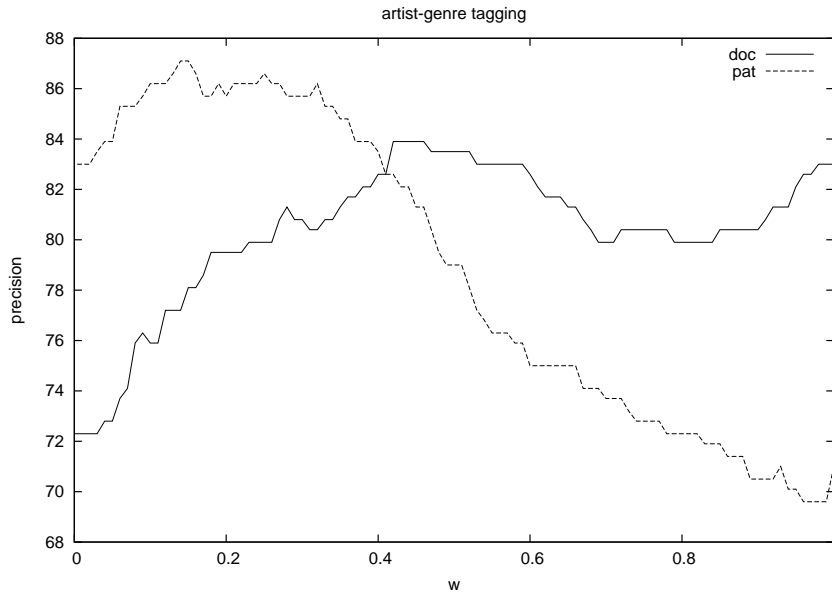


Figure 2: Precision (%) for best scoring tags.

we used artists outside the test set. For the similarity between the artists in A , the patterns found were mostly enumeration patterns, e.g. “including [artist] and [artist]”. We used seven of them. Using DOC, we downloaded the first 100 documents retrieved per query.

METHOD	NUMBER OF NEAREST NEIGHBORS CONSIDERED						
	$n = 1$	$n = 2$	$n = 3$	$n = 5$	$n = 7$	$n = 10$	$n = 15$
PAT	89.4	83.2	79.0	72.3	67.0	61.0	53.6
DOC	69.2	64.6	60.1	54.5	47.9	39.4	28.0

Table 1: Average precision of the n nearest artists sharing the same genre.

We test the precision of the artist similarity method using the genres in the evaluation set. If two artists in this set share the same genre, we consider them closely related. Using the scores $t(a, b)$, for all artists we computed an ordered list of similar artists. Per artist a , we checked whether the n nearest neighbors, based on the scores $t(a, b)$, share the same genre in the evaluation set. In Table 1 the average percentages of nearest neighbors can be found that share the same genre with artist a .

Using the co-occurrences found with PAT, the precision for one nearest neighbor is 89.4%. As can be expected, the precision for PAT decreases when considering more neighbors. However, considering that per genre only 16 artists are identified for $n = 15$ the precision still is 53.6%. The performance of DOC is less precise. For larger numbers of n the precision drops. This can be explained by the fact that considerably smaller numbers of co-occurrences are found using this method. For some artists, co-occurrences could be found with only five other artists. The pattern-based method PAT thus performs best in precision and recall with respect to artist similarity.

In Table 2 and Figure 2, the precision of the genre tagging method can be found. Per artist, the best scoring genre using $p(a, l)$ is evaluated using the evaluation set. Note that we only check for precision, and not for recall. If no tags could be found for an artist, we considered the result as erroneous.

When no similar artists are taken into account (hence $w = 1$), DOC outperforms PAT. How-

METHOD	ARTIST-GENRE TAGGING			
	$w = 1$	$w = 0$	best	(corresp. w)
PAT	71.0	83.0	87.1	0.14
DOC	81.0	72.3	83.9	0.43

Table 2: Precision (%) for best and extreme values for w .

ever, since the artist similarity scores using PAT are more reliable than those of DOC (Table 1), the performance of PAT increases with the weight of the neighboring artists. For DOC however, small values of w have a negative effect on the precision. The best results for both methods are nevertheless obtained using a w smaller than 1. We thus can conclude that incorporating similarities between artists improves the performance in this experiment.

With the supervised music artist clustering method discussed in (Knees *et al.*, 2004) a precision of 87% was obtained using complex machine learning techniques and a relatively large training set. In (Schedl *et al.*, 2005a) a precision of up to 85% precision was obtained using $O(|A|^2)$ Google queries. We can conclude that our simple and unsupervised method produces similar results. Moreover, we explicitly tag artists with genres instead of computing clusters of artists. Recent work by Schedl *et al.* (2006) does focus on the direct tagging of artists. The method described in this work uses $O(|A| \cdot |L|)$ Google Complexity and uses a scoring function alike $s(a, l)$ to select the most appropriate tag. They report a precision of 71% on a test set with only 9 distinct genres.

7.2 Tagging painters with art-styles

In this experiment we are interested in whether our method is also applicable to a different domain. We map a set of painters A to a set of styles and movements in art (L). From Wikipedia we extracted a list of 1,280 well-known painters from the article *List of painters* and a list of 77 art-styles from *List of art movements*². We tested the performance of the algorithm on the subset of 160 painters who could be extracted from pages describing styles (e.g. from the page on *Abstract Expressionism*). The other 1,120 painters are either not mentioned on the pages describing styles or are mentioned on more than one page. However, when computing similarities between the painters, we take all 1,280 painters into account.

"[painter] [movement]"
"[movement] [painter]"
"[painter] and other [movement]"
"[painter] and [movement]"
"[painter] tog initiativ til [movement]"
"[painter] and the [movement]"
"[painter] surrealism [movement]"
"[painter] synthetic [movement]"
"[movement] artist [painter]"
"[painter] express [movement]"
"[painter] of the [movement]"
"[painter] uit de [movement]"
"[painter] experimenting with [movement]"
"[painter] arte [movement]"
"[movement] painter [painter]"

Table 3: Best scoring learned patterns for painter - movement relation.

²www.wikipedia.org Both pages visited in April 2006.

For the elements of L in this test no synonyms were added. For fairness, we excluded pages from the domain `wikipedia.org` in the Google search queries.

In Table 3 we give the best scoring learned patterns for the painter-movement relation (used in PAT). For the relation between pairs of artists, we used the same enumeration patterns as in the first experiment.

In this experiment we do not have a ground truth for the best applicable movement to every artist. We therefore cannot evaluate the performance of the artist similarity scores.

The results of the movement tagging experiment are given in Table 4 and Figure 3. Although in the painter-movement experiment the number of categories (77) is much larger than in the first one (14), the performance of PAT and especially DOC is still good for smaller values of w .

It is notable that in this experiment the precision of both methods improves when increasing the effect of the neighboring painters (i.e. by decreasing w). The precision for $w = 0$ (the direct scoring $s(a, l)$ not taken into account) is thus better than the performance of $s(a, l)$ (i.e. $p(a, l)$ for $w = 1$). Best results are obtained for both PAT and DOC for values of w around 0.15.

Although the performance of the second experiment is less good than the one of the first experiment, we can conclude that the results of the painters tagging task are convincing given the size of the set of tags L .

PAINTER-MOVEMENT TAGGING				
METHOD	$w = 1$	$w = 0$	best	(corresp. w)
PAT	48.1	55.6	63.8	0.05
DOC	58.8	66.3	78.8	0.18

Table 4: Precision (%) for best and extreme values for w .

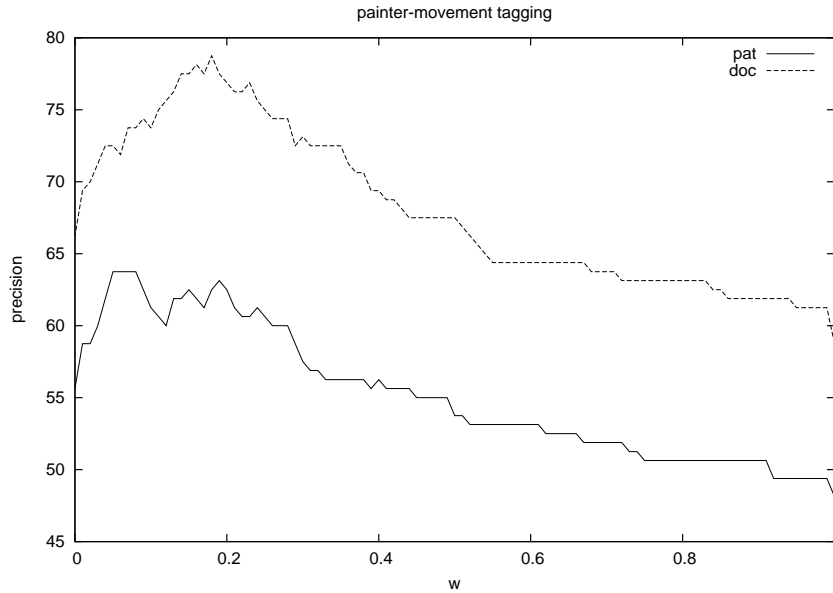


Figure 3: Precision (%) for best scoring tags.

8 Conclusions and Future Work

We have discussed two efficient alternative methods (PAT and DOC) to obtain co-occurrences of terms using a web search engine. These methods are applied to gain scoring functions for mutual relatedness between artists (or painters) and for relatedness between artists and tags (genres or art-styles). Using the assumption that the same tags are applicable to similar artists, we combine these scoring functions into a final score. We use a weight w for the initial scores between artists and tags.

The method discussed to obtain the co-occurrences are linear in the number of items in the sets of artists and tags, where alternative approaches in literature are quadratic. This distinction is important for tagging large sets of artists, since search engines allow only a limited amount of automated queries per day.

We can precisely find similar artists, especially using PAT. Both approaches lead to good results when tagging artists with genres. A second experiment consisted of the tagging of painters with their art-styles. Here, the DOC method gives good results. However, it is not possible to identify a uniform reliable value for w based on these experiments.

The results of the two experiments are encouraging. Using the computed similarities between artists indeed helps to improve the tagging of the artists. The experiments show that taking similar artists into account improves the performance of the tagging.

In future work, we do not only want to evaluate the best scoring tag, but rather the best n tags. We want to compose a more diverse set L with tags commonly used in folksonomies such as last.fm. Using a larger set of artists, we want to evaluate the performance for the n best scoring tags per artists using data from such a large tagging community.

Moreover, currently we assume the tags in the set L to be given. We are interested to exploit methods to learn new terms for the set L . This can for instance be done with the *tf-idf*-approach (Knees *et al.*, 2004; Manning & Schütze, 1999).

References

- Agichtein, E., & Gravano, L. 2000. Snowball: Extracting Relations from Large Plain-Text Collections. *In: Proceedings of the Fifth ACM International Conference on Digital Libraries*.
- Boer, V. de, Someren, M. van, & Wielinga, B. J. 2006. Extracting Instances of Relations from Web Documents using Redundancy. *In: Proceedings of the Third European Semantic Web Conference (ESWC'06)*.
- Brin, S. 1998. Extracting Patterns and Relations from the World Wide Web. *In: WebDB Workshop at sixth International Conference on Extending Database Technology (EDBT'98)*.
- Brin, S., & Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, **30**(1-7), 107-117.
- Brooks, C.H., & Montanez, N. 2006. Improved annotation of the blogosphere via autotagging and hierarchical clustering. *Pages 625-632 of: Proceedings of the 15th international conference on World Wide Web (WWW2006)*.
- Cilibrasi, R., & Vitanyi, P. 2004. *Automatic Meaning Discovery Using Google*. <http://www.cwi.nl/~paulv/papers/amdug.pdf>.
- Cimiano, P., & Staab, S. 2004. Learning by Googling. *SIGKDD Explorations Newsletter*, **6**(2), 24-33.
- Crescenzi, V., & Mecca, G. 2004. Automatic information extraction from large websites. *Journal of the ACM*, **51**(5), 731-779.
- Downey, D., Etzioni, O., & Soderland, S. 2005. A Probabilistic Model of Redundancy in Information Extraction. *Pages 1034-1041 of: 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*.
- Etzioni, O., Cafarella, M. J., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D. S., & Yates, A. 2005. Unsupervised named-entity extraction from the Web: An experimental study. *Artificial Intelligence*, **165**(1), 91-134.

- Geleijnse, G., & Korst, J. 2006a. Learning Effective Surface Text Patterns for Information Extraction. *Pages 1–8 of: Proceedings of the EACL 2006 workshop on Adaptive Text Extraction and Mining (ATEM 2006)*.
- Geleijnse, G., & Korst, J. 2006b. Web-based Artist Categorization. *Pages 266 – 271 of: Proceedings of the Seventh International Conference on Music Information Retrieval (ISMIR'06)*.
- Geleijnse, G., Korst, J., & de Boer, V. 2006. Instance Classification using Co-occurrences on the Web. *In: Proceedings of the ISWC 2006 workshop on Web Content Mining with Human Language Technologies*.
- Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. *Pages 539–545 of: Proceedings of the 14th conference on Computational linguistics*.
- Knees, P., Pampalk, E., & Widmer, G. 2004. Artist Classification with Web-based Data. *Pages 517–524 of: Proceedings of 5th International Conference on Music Information Retrieval (ISMIR'04)*.
- Korst, J., Geleijnse, G., de Jong, N., & Verschoor, M. 2006. Ontology-Based Extraction of Information from the World Wide Web. *Pages 149 – 167 of: Intelligent Algorithms in Ambient and Biomedical Computing*. Philips Research Book Series. Springer.
- Manning, C.D., & Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press.
- O'Reilly, T. 2005. *What is Web2.0*. <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.
- Ravichandran, D., & Hovy, E. 2002. Learning surface text patterns for a Question Answering System. *Pages 41–47 of: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*.
- Schedl, M., Knees, P., & Widmer, G. 2005a. A Web-Based Approach to Assessing Artist Similarity using Co-Occurrences. *In: Proceedings of the Fourth International Workshop on Content-Based Multimedia Indexing (CBMI'05)*.
- Schedl, M., Knees, P., & Widmer, G. 2005b. Discovering and Visualizing Prototypical Artists by Web-based Co-Occurrence Analysis. *In: Proceedings of the Sixth International Conference on Music Information Retrieval (ISMIR'05)*.
- Schedl, M., Pohle, T., Knees, P., & Widmer, G. 2006. Assigning and Visualizing Music Genres by Web-based Co-Occurrence Analysis. *In: Proceedings of the Seventh International Conference on Music Information Retrieval (ISMIR'06)*.
- Véronis, J. 2006. *Weblog*. <http://aixtal.blogspot.com>.
- Zadel, M., & Fujinaga, I. 2004. Web Services for Music Information Retrieval. *In: Proceedings of 5th International Conference on Music Information Retrieval (ISMIR'04)*.