# Automatic Computation of Semantic Proximity Using Taxonomic Knowledge

Cai-Nicolas Ziegler*        Kai Simon        Georg Lausen

DBIS, Institut für Informatik, Universität Freiburg
Georges-Köhler-Allee, Gebäude 51
79110 Freiburg i.Br., Germany

{cziegler,ksimon,lausen}@informatik.uni-freiburg.de

## ABSTRACT

Taxonomic measures of semantic proximity allow us to compute the relatedness of two concepts. These metrics are versatile instruments required for diverse applications, e.g., the Semantic Web, linguistics, and also text mining. However, most approaches are only geared towards hand-crafted taxonomic dictionaries such as WORDNET, which only feature a limited fraction of real-world concepts. More specific concepts, and particularly *instances* of concepts, i.e., names of artists, locations, brand names, etc., are not covered.

The contributions of this paper are twofold. First, we introduce a framework based on Google and the Open Directory Project (ODP), enabling us to derive the sematic proximity between *arbitrary* concepts and instances. Second, we introduce a new taxonomy-driven proximity metric tailored for our framework. Studies with human subjects corroborate our hypothesis that our new metric outperforms benchmark semantic proximity metrics and comes close to human judgement.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Retrieval and Search—*Information Filtering*; I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge Acquisition*

## General Terms

Algorithms, Experimentation, Human Factors, Measurement

## Keywords

Semantic similarity, metrics, taxonomy, accuracy, data extraction

------

*Now working for Siemens AG, Corporate Technology IC 1, Munich. Contact through indicated e-mail.

## 1. INTRODUCTION

Research on similarity of word meanings dates back to the early 60's [17]. Thenceforward, numerous semantic proximity measures have been proposed, mostly operating on the taxonomic dictionary WORDNET [13, 15, 11, 3, 10], exploiting its hierarchical structuring. The main objective of these approaches is to mimic human judgement with respect to the relatedness of two concepts, e.g., BICYCLES and CARS. With the advent of applications that intend to make machines understand and extract *meaning* from human-crafted information, e.g., the Semantic Web initiative or text mining, the necessity for tools enabling the automatic detection of semantic proximity becomes even stronger. However, one severe drawback of these approaches is that their application has been confined to WORDNET only. While the number of unique strings of this taxonomically organized dictionary, i.e., nouns, verbs, adjectives, and adverbs, nears 150,000, large amounts of information available on the Web and other textual sources cannot be captured by such dictionaries. Particularly brand names, names of artists, locations, products and composed terms, in other words, specific *instances* of concepts, are beyond their scope. Examples are CIKM, DATABASE THEORY, NEIL ARMSTRONG, or XBOX GAMES, to name some.

### 1.1 Contributions

We intend to overcome the aforementioned issue and propose an architecture that allows to compute the estimated semantic proximity between *arbitrary* concepts and concept instances.[1] The following two major contributions are made:

- **Framework leveraging Google and ODP.** Instead of merely exploiting WORDNET, we leverage the power of the entire Web and the endeavors of thousands of voluntary human editors who classify Web pages into the ODP taxonomy. Google serves as an instrument to obtain Web pages that match one particular concept, e.g., FRIEDRICH SCHILLER. The Open Directory Project (ODP) then allows us to classify these result pages into a human-crafted taxonomy. Thus, we are able to garner a *semantic profile* of each concept.

- **Metric for multi-class categorization.** Most tax-

------

[1]We will abuse language by likewise denoting *concepts*, e.g., POET, and *instances*, e.g., FRIEDRICH SCHILLER, by the term *concept* only.
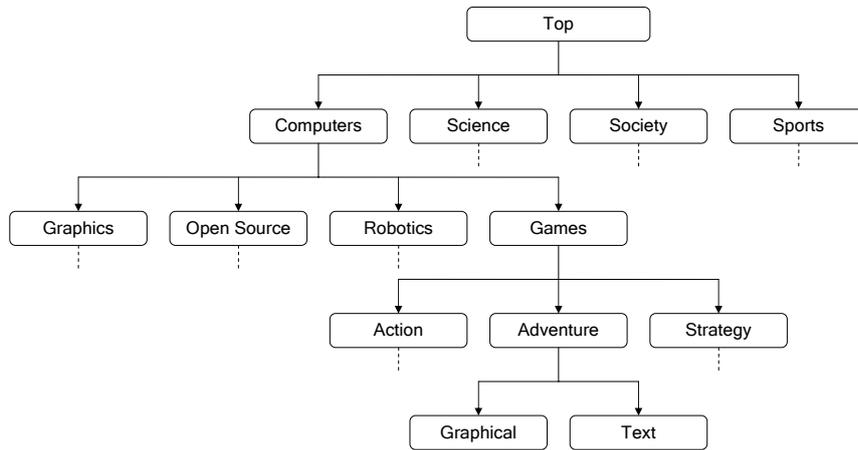
**Figure 1: Extracted fragment from the ODP taxonomy**

onomy-based proximity measures are geared towards computing the semantic distance of word senses that fall into exactly *one* category each. We propose a metric that allows a multi-class approach [22] where each given concept may be arranged into several categories, which is essential for our architectural setup. Empirical evaluation through an online user study demonstrates the superior performance of our approach against traditional similarity metrics.

## 1.2 Paper Organization

Our work is structured as follows. In Section 2, we survey relevant existing literature on semantic similarity. Next, we describe our system's architectural framework based on Google and ODP. Section 4 presents some taxonomy-based proximity measures and introduces our new proximity metric. Amalgamating our system architecture with such metrics, we conduct an extensive empirical evaluation in Section 5, involving more than 50 human subjects. Eventually, we give an outlook of future research, taking the notion of semantic proximity one step further.

## 2. RELATED WORK

The study of semantic proximity between two given concepts has largely focused on *similarity*, e.g., synonymy and hyponymy [13]. Proximity goes even further, also subsuming meronymy (part-whole) and arbitrarily typed semantic relationships.

Early taxonomy-based similarity metrics only have taken into account the shortest path $\phi$ between two concepts within the taxonomy, and the depth $\sigma$ of the most specific common subsumer of both concepts. See [3] for an overview of these early works. Next-generation approaches were inspired by information theory and only used taxonomies in combination with text corpora. Thus, the probability of a concept, its so-called *information content*, could be computed and used to refine the similarity measure. Resnik [15, 16] lay the foundations of this approach, followed by Jiang and Conrath [9] and Lin [11], both largely similar to Resnik's. Lin's approach is extended to handle graphs rather than mere trees by Maguitman *et al.* [12].

Li *et al.* [10] have conducted an extensive study that revealed that the usage of information content does not yield better performance. Moreover, they proposed a metric that combines shortest path length $\phi$ and subsumer depth $\sigma$ in a *non-linear* fashion and outperformed traditional taxonomy-based approaches. Chirita *et al.* [5] used a variation of Li *et al.*'s metric for personalizing Web search.

Taxonomy-based metrics with collaborative filtering systems in mind have been proposed by Ganesan *et al.* [8] and Ziegler *et al.* [22].

The exploitation of Web search engine results for computing similarity between concepts or, more generally, queries, has been tried before. Chien and Immorlica [4] and Vlachos *et al.* [20] attempted to detect similarity via temporal correlation, reporting mixed results. Wen *et al.* [21] computed semantic query similarity based on query content, user feedback, and some simple document hierarchy. No empirical analyses were provided, though. Cimiano *et. al.* [6, 7] make use of linguistic patterns along with search engine leverage to automatically identify class-instance relationships and disambiguate word senses.

## 3. FRAMEWORK

In this section, we describe the framework we built in order to compute semantic proximity between arbitrary concepts or instances. The approach rests upon ODP and Google Directory, which we use in order to provide us with the required background knowledge. The two services are required so we can compose *semantic profiles* of the concepts we want to compare. The following step then necessitates a proximity metric to match these profiles against each other.

### 3.1 Service Requirements

**Open Directory Project.** The so-called DMOZ Open Directory Project (*http://www.dmoz.org*) combines the joint efforts of more than 69,000 volunteering editors helping to categorize the Web. ODP is the largest and most comprehensive human-edited Web page catalog currently available. Organized as a tree-structured taxonomy, 590,000 categories build the inner nodes of the directory. Leaf nodes are given by more than 5.1 mil-

lion sites that have been categorized into the ODP already.

**Google Directory.** Building upon ODP, Google Directory (*http://www.google.com/dirhp*) provides search results with additional *referrals* into the ODP, thus extending the traditional Google service. These referrals represent paths from the ODP's inner nodes or leafs to its root node, $\top$. For one given query, only those pages are returned as results that have been categorized into the ODP. For example, the ODP referral that Google Directory's first search result assigns to the ACM main page (*http://www.acm.org*) looks as follows:

$$\top \rightarrow \textsc{Computers} \rightarrow \textsc{Comp. Sc.} \rightarrow \textsc{Organizations} \ldots \quad (1)$$

## 3.2 Algorithm Outline

The main task is to compute the proximity $s$ between two concepts $c_x$, $c_y$, i.e., $s(c_x, c_y)$. In order to match $c_x$ against $c_y$, we need to build semantic profiles for both concepts first.

To this end, we send these two concepts $c_x$ and $c_y$, for instance $\textsc{Boris Becker}$ and $\textsc{Wimbledon}$, to Google Directory and obtain two *ranked result lists* $q^{c_x} : \mathcal{L}^{n_x}$ and $q^{c_y} : \mathcal{L}^{n_y}$, respectively. We define $\mathcal{L}^{n_z} := \{1, 2, \ldots, n_z\} \rightarrow D$, where $z \in \{x, y\}$, $n_z$ the number of documents returned for query term $c_z$, and $D$ the set of topics in the ODP taxonomy. Hence, we only consider the *ODP referral* associated with each document returned for the query rather than the textual summary. For example, $q^{c_x}(1)$ gives the ODP topic that the top-ranked result document for query term $c_x$ is categorized into.

Next, the two profiles are forwarded to the proximity metric, which then computes the estimated semantic proximity score $s(c_x, c_y)$, using the ODP as background knowledge to look up topics $q^{c_z}(i)$, where $z \in \{x, y\}$ and $i \in \{1, 2, \ldots, n_z\}$, within the taxonomy.

The approach is very versatile and not only extends to the computation of semantic proximity for pairs of concepts, but effectively pairs of arbitrary queries in general.

## 4. PROXIMITY METRICS

In order to use our framework, we need to install an actual taxonomy-based proximity metric $s$ that compares $q^{c_x}$ with $q^{c_y}$. Since $q^{c_x}$ and $q^{c_y}$ are ranked lists of topics, its type must be $s : (\mathcal{L}^{n_x} \times \mathcal{L}^{n_y}) \rightarrow S$, where $S$ is an arbitrary scale, e.g., $[-1, +1]$.

We will first review several popular metrics that have been proposed in the context of $\textsc{WordNet}$ and then proceed to propose our own taxonomy-based proximity metric.

## 4.1 Similarity Between Word-Sense Pairs

In general, $\textsc{WordNet}$ metrics compare the similarity of *word senses*, where each word sense is represented by a topic from the taxonomy. Hence, the metric's functional layout is $s : D \times D \rightarrow S$ rather than $s : (\mathcal{L}^{n_x} \times \mathcal{L}^{n_y}) \rightarrow S$. The reason is that we are comparing two *topics* from the taxonomy rather than *instances*, which are arranged into multiple topics within the taxonomy. For example, $\textsc{Batman}$ is an instance of *several* topics, e.g. $\textsc{Movie}$ and $\textsc{Cartoon Hero}$. We will show later on how to circumvent this issue.

The simplest $\textsc{WordNet}$ metric only computes the shortest path $\phi(d_x, d_y)$ between two topics $d_x, d_y$ in the taxonomy. Leacock and Chodorow [3] modify this basic metric by scaling the path length by the overall depth $\lambda$ of the taxonomy:

$$s_{LC}(d_x, d_y) = -\log\left(\frac{\phi(c_x, c_y)}{2 \cdot \lambda}\right) \quad (2)$$

Though the Leacock-Chodorow metric uses little information to compute the similarity estimate, its accuracy has been shown only insignificantly inferior to more informed approaches based on information theory, e.g., Resnik [15, 16] or Jiang and Conrath [9].

Li *et al.* [10] have conducted an extensive survey comparing numerous existing $\textsc{WordNet}$ metrics and proposed a new metric which combines shortest path length $\phi(d_x, d_y)$ and most specific subsumer depth $\sigma(d_x, d_y)$ in a *non-linear* fashion, outperforming all other benchmark metrics. The most specific subsumer of topics $d_x$ and $d_y$ is defined as the topic that lies on the taxonomy paths from the root topic to both $d_x$ and $d_y$ and has maximal distance from the root topic. Li *et al.*'s metric also features two tuning parameters $\alpha$ and $\beta$ which have to be learned in order to guarantee the metric's optimal performance. The metrics is defined as follows:

$$s_{LBM}(d_x, d_y) = e^{-\alpha \cdot \phi(d_x, d_y)} \cdot \frac{e^{\beta \cdot \sigma(d_x, d_y)} - e^{-\beta \cdot \sigma(d_x, d_y)}}{e^{\beta \cdot \sigma(d_x, d_y)} + e^{-\beta \cdot \sigma(d_x, d_y)}} \quad (3)$$

As has been stated before, the above metrics only measure the distance between two singleton topics $d_x$ and $d_y$ rather than two lists of topics $q^{c_x} : \mathcal{L}^{n_x}$ and $q^{c_y} : \mathcal{L}^{n_y}$, respectively. The issue has been addressed [5] by computing the average similarity of all unordered pairs $\{d_x, d_y\} \in \Im(q^{c_x}) \times \Im(q^{c_y})$, where $d_x \neq d_y$.[2]

## 4.2 Multi-Class Categorization Approach

As has been outlined in Section 3.2, multi-class categorization of concepts/instances into *several* topics is essential for our approach, since more than one query result and its taxonomic referral are used to describe one concept/instance $c_z$. Existing $\textsc{WordNet}$ metrics can be tailored to support proximity computations for topic lists, but their performance is non-optimal (see Section 5). We therefore propose a new metric with multi-class categorization in mind. Our approach is substantially different from existing metrics and can be subdivided into two major phases, namely *profiling* and *proximity computation*. The first phase takes the list of topics describing one concept/instance $c_z$, e.g., $\textsc{Batman Begins}$, and creates a flat profile vector, based on the ODP taxonomy as background knowledge. The second phase then takes the profile vectors for both $c_x$ and $c_y$ and matches them against each other, hence computing their correlation.

As input, our metric expects two ranked topics lists $q^{c_z} : \mathcal{L}^{n_z}, z \in \{x, y\}$, and three fine-tuning parameters, $\alpha$, $\gamma$, and $\delta$. These parameters have to be learned from a training set before applying the metric.

### 4.2.1 Profiling Phase

For each concept $c_z$ for which to build its profile, we create a new vector $\vec{v}_z \in \mathbb{R}^{|D|}$, i.e., the vector's dimension is exactly the number of topics in the ODP taxonomy. Next, we accord a certain score $\tilde{\mu}_i$, where $i \in \{1, 2, \ldots, n_z\}$, for each topic $q^{c_z}(i) \in \Im(q_z)$. The amount of score depends on the *rank* $i$ of topic $q^{c_z}(i)$. The earlier the topic appears in

---

[2] $\Im(f)$ denotes the *image* of map $f : A \rightarrow B$, i.e., $\Im(f : A \rightarrow B) := \{f(x) \mid x \in A\}$.

the result list $q^{c_z}$ of query $c_z$, the more weight we assign to that topic, based upon the assumption that results further down the list are not as valuable as top-list entries. For the weight assignment, we assume an *exponential decay*, inspired by Breese *et al.*'s half-life utility metric [2]:

$$\tilde{\mu}_i = 2^{-(i-1)/(\alpha-1)} \qquad (4)$$

Parameter $\alpha$ denotes the *impact weight half-life*, i.e., the number of the rank of topic $q^{c_z}(\alpha)$ on list $q^{c_z}$ for which the impact weight is exactly half as much as the weight $\tilde{\mu}_1$ of the top-ranked result topic. When assuming $\alpha = \infty$, all ranks are given equal weight.

Having computed the score $\tilde{\mu}_i$ for each topic $q^{c_z}(i)$, we now start to assign score to all topics $d_{i,0}, d_{i,1}, \ldots, d_{i,\lambda(i)}$ along the path from $q^{c_z}(i)$ to the taxonomy's root node. Hereby, $\lambda(i)$ denotes the *depth* of topic $q^{c_z}(i) = d_{i,\lambda(i)}$ in our taxonomy, and $d_{i,0}$ is the root node. The idea is to propagate score from leaf topic $d_{i,\lambda(i)}$ to all its ancestors, for each $d_{i,j}$ is also "a type of" $d_{i,j-1}$, owing to the taxonomy's nature of being composed of hierarchical "is-a" relationships.

Note that the ODP taxonomy also features some few links of types other than "is-a", namely "symbolic" and "related". These types were not considered in our model so far. When upward-propagating score for each $q^{c_z}(i)$, we first assign score $\mu_{i,\lambda(i)} := \tilde{\mu}_i$ to $d_{i,\lambda(i)}$. The score for its parent topic $d_{i,\lambda(i)-1}$ then depends on four factors, namely parameters $\gamma$ and $\delta$, the number of siblings of $d_{i,\lambda(i)}$, denoted $\rho(d_{i,\lambda(i)})$, and the score $\mu_{i,\lambda(i)}$ of $d_{i,\lambda(i)}$. The general score propagation function from taxonomy level $j$ to level $j-1$ is given as follows:

$$\mu_{i,j-1} = \mu_{i,j} \cdot \frac{1}{\gamma + \delta \cdot \log(\rho(d_{i,j}) + 1)} \qquad (5)$$

Informally, the propagated score depends on a constant factor $\gamma$ and the number of siblings that topic $d_{i,j}$ has. The more siblings, the less score is propagated upwards. In order to not overly penalize nodes $d_{i,j-1}$ that have numerous children, we chose *logarithmic* scaling. Parameter $\delta$ controls the influence that the number of siblings has on upward propagation. Clearly, other functions could be used likewise.

Next, we normalize all score $\mu_{i,j}$, where $i \in \{1, 2, \ldots, n_z\}$ and $j \in \{0, 1, \ldots, \lambda(i)\}$, so that values $\mu_{i,j}$ sum up to unit score. Values of vector $\vec{v}_z$ at positions $d_{i,j}$ are eventually increased by $\mu_{i,j}$, yielding the final *profile vector* for concept $c_z$.

Algorithm 1 summarizes the complete profiling procedure. Function $\text{pathvec}(q^{c_z}(i) \in D)$ returns the vector containing the path of topic $q^{c_z}(i)$'s ancestors, from $q^{c_z}(i)$ itself to the root node. The vector's size is $\lambda(i)$. Function $\text{id}(d \in D)$ gives the *index* that topic $d$ is mapped to within profile vector $\vec{v}_z$.

### 4.2.2 Measuring Proximity

Profile generation for concepts $c_x, c_y$ and their respective ranked topic lists $q^{c_x}, q^{c_y}$ appears as the major task of our approach; the eventual proximity computation is straightforward. Mind that the profiling procedure generates *plain feature vectors*, so we can apply generic statistical tools for measuring vector similarity. We opted for Pearson's correlation coefficient, particularly prominent in collaborative filtering applications [18, 23]. Hence, the final proximity value is computed as follows:

```
func prof (q^{c_z} : L^{n_z}, α, γ, δ) returns v⃗_z ∈ ℝ^{|D|} {
    set v⃗_z ← 0⃗, n ← 0;
    for i ← 1 to n_z do
        set μ̃_i ← 2^{-(i-1)/(α-1)};
        for j ← λ(i) to 0 do
            set d_{i,j} ← ( pathvec(q^{c_z}(i)) )_j;
            if ( j = λ(i) ) then
                set μ_{i,j} ← μ̃_i;
            else
                set μ_{i,j} ← μ_{i,j+1} · (γ + δ · log( ρ(d_{i,j} + 1) )^{-1};
            end if
            set v_{z,id(d_{i,j})} ← v_{z,id(d_{i,j})} + μ_{i,j};
            set n ← n + μ_{i,j};
        end do
    end do
    for i ← 1 to |D| do
        set v_{z,i} ← v_{z,i} / n;
    end do
    return v⃗_z;
}
```

**Algorithm 1: Profiling algorithm**

$$s_{ZSL}(\vec{v}_x, \vec{v}_y) = \frac{\sum\limits_{k=0}^{|D|} (v_{x,k} - \overline{v}_x) \cdot (v_{y,k} - \overline{v}_y)}{\left( \sum\limits_{k=0}^{|D|} (v_{x,k} - \overline{v}_x)^2 \cdot \sum\limits_{k=0}^{|D|} (v_{y,k} - \overline{v}_y)^2 \right)^{\frac{1}{2}}} \qquad (6)$$

Where $\overline{v}_x$ and $\overline{v}_y$ denote the mean values of vectors $\vec{v}_x$ and $\vec{v}_y$. Moreover, $\vec{v}_x$ and $\vec{v}_y$ are assumed to have been computed according to Algorithm 1.

## 5. EMPIRICAL EVALUATION

In Section 3, we have proposed a framework to compute semantic proximity between *arbitrary* concepts/instances. In order to evaluate which proximity metric best fits our approach, we conducted an extensive empirical study involving 51 human subjects and necessitating the creation of two novel benchmark sets, featuring 30 and 25 concept pairs.

The evaluation method follows the methodology used for comparing the performance of WORDNET metrics, e.g., [15], [11], [3], and [10], based on mainly two benchmark sets, namely Rubenstein-Goodenough [17] and Miller-Charles [14]. The first set features 65 concept pairs, e.g., ROOSTER vs. VOYAGE, FURNACE vs. STOVE, and so forth. Miller-Charles is a mere subset of Rubenstein-Goodenough and only contains 30 word pairs. These 30 word pairs were given to a group of 38 people, asking them to judge the semantic similarity of each pair of words on a 5-point scale [14]. For benchmarking, these human ratings were used as an "arbiter" for all WORDNET metrics, and the correlation for each metric's computed word/concept pair similarities with human judgement was measured. The closer the metric's results, the better its accuracy.

## 5.1 Benchmark Layout

Neither Miller-Charles' nor Goodenough-Rubenstein's set features concept instances or composed concepts, e.g., locations, book titles, names of actors, etc.; however, the comparison of semantic proximity for these specific terms represents the core capability of our framework. We hence needed to create own benchmark sets. Since some of the metrics presented in Section 4, namely Li *et al.*'s [10] metric and our own approach, require parameter learning, we created *two* lists of concept pairs. The first, denoted $B_0$, contains 25 pairs of concepts, e.g., FOOD NETWORK vs. FLOWERS, or IMDB vs. BLOCKBUSTER, and serves for training and parameter learning. The second benchmark, denoted $B_1$, features 30 concept pairs such as EASYJET vs. CHEAP FLIGHTS and HOLIDAY INN vs. VALENTINE'S DAY. The conception of both benchmark sets and their respective human subject studies is identical. They only vary in their concept pair lists, which are disjoint from each other. The sets, along with the average ratings of human subjects per concept pair and the respective standard deviations, are given in Table 1 and 2.

In order to obtain these two lists of concept pairs, we used Google's Suggest service, still in its beta version at the time of this writing (*http://www.google.com/webhp?complete=1*): For each letter in the alphabet (A-Z), we collected the list of most popular search queries proposed by Google Suggest, giving us 260 different queries, e.g., CHEAP FLIGHTS, INLAND REVENUE, and CARS. We took all 33,670 possible query pair combinations and computed the semantic proximity for each query according to our proposed framework, using the simple Leacock-Chodorow [3] metric. Next, for the test set $B_1$, we sorted all pairs according to their metric score in descending order and randomly picked ten pairs from the 5% most similar concept pairs, ten concept pairs from midrange, and ten pairs from the bottom. For $B_0$, we proceeded in a similar fashion.

We still had to manually weed out unsuitable pairs, i.e., those terms that people were deemed to be unfamiliar with. For instance, the famous US series DESPERATE HOUSEWIVES is largely unknown to Germans, who represented 87% of all participants.

Though comparatively laborious, we opted for the largely automatized and randomized method presented above rather than for manual selection, which might have incurred personal bias into the design of both benchmark sets.

## 5.2 Online Survey Design

Both online studies exhibited an identical make-up, differing only in the concept pairs contained. Participants were required to rate the semantic relatedness of *all* concept pairs on a 5-point likert scale, ranging from *no proximity* to *synonymy*. 51 people completed $B_1$ and 23 of them also filled out $B_0$.[3] 87% of $B_1$'s participants were German, while the remaining 13% were Italian, Turkish, US-American, and Israeli. 27% of all participants were CS PhD students or faculty, much lower than for Resnik's replication of Miller-Charles.

The results of each survey $B_z$, $z \in \{0, 1\}$, were regarded as the proximity rating vector $\vec{v}_i \in \{1, 2, \ldots, 5\}^{|B_z|}$ for the respective participant $i$. We thus computed the *inter-subject correlation*, i.e., the Pearson correlation coefficient $p(\vec{v}_i, \vec{v}_j)$

---

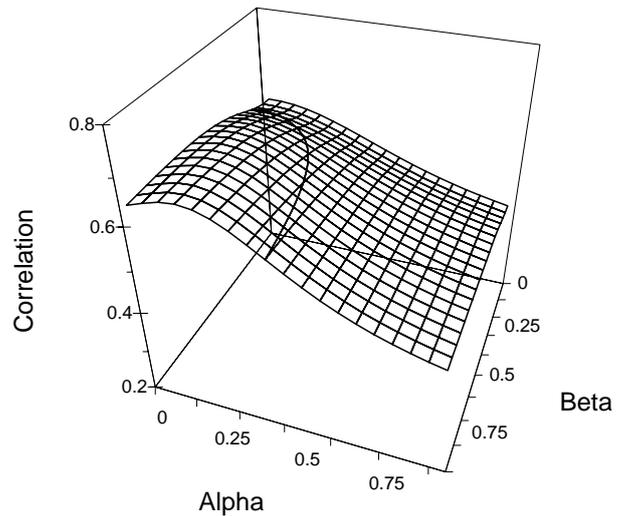[3]Owing to $B_1$'s major importance, we asked people to complete $B_1$ first.



**Figure 2: Parameter learning curve for Li *et al.***

(see Section 4.2.2) for every unordered pair $\{i, j\} \in B_z \times B_z$ of human subjects, represented by their proximity rating vectors. Pair similarity scores were summed up and averaged afterwards:

$$p_z = \sum_{\{i,j\} \, \in \, (B_z \times B_z)} p(\vec{v}_i, \vec{v}_j) \cdot \frac{2}{|B_z| \cdot (|B_z| - 1)} \qquad (7)$$

For set $B_1$, we obtained an average correlation $p_1 = 0.7091$. For $B_0$, we had an average correlation $p_0$ of 0.7027. These values bear strong indication for judgement correlation, but are still considerably lower than Resnik's human judgment replication of Miller-Charles, which had an inter-subject correlation of 0.8848 [15].

We identify the following points as driving forces behind this observation:

- **Instances versus concepts.** Miller-Charles only featured concepts contained in dictionaries. We rather focused on names of artists, brand names, composed and qualified concepts. These are harder to handle as they are more specific and require a certain extent of domain knowledge (e.g., knowing that GMAIL is Google's electronic mail service).

- **Language and demographics.** In Resnik's replication [15], ten people affiliated with CS research participated. However, in our experiment, demographics were much more wide-spread and CS researchers only had an overall share of 27% for $B_1$. Even larger perturbation was caused by language, for most of the participants were German-speaking while the study itself was in English.

## 5.3 Proximity Metrics

For measuring proximity, we compared several strategies which can be categorized into two larger classes, namely *taxonomy-based* (see Section 4) and *text-based*. While focusing on the group of taxonomy-based metrics, the second group served as an indication to verify that traditional text-based

methods cannot provide better performance, thus rendering the new breed of taxonomy-based metrics obsolete for our purposes.

### 5.3.1 Taxonomy-driven Metrics

For the taxonomy-based category, we opted for the metrics presented in Section 4, namely Leacock-Chodorow [3], Li *et al.* [10], and our own approach, henceforth "Ziegler *et al.*". For both WORDNET metrics, i.e., Li *et al.* and Leacock-Chodorow, we employed the strategy presented in Section 4.1 for comparing *concept lists* rather than singletons.

### 5.3.2 Text-based Approaches

Besides taxonomy-driven metrics, we also tested various classic *text-based* similarity measures for achieving the task, based upon the well-known vector-space paradigm [19, 1]. To this end, we tested four setups, using two different types of data to run upon:

First, instead of using the taxonomic description $q^{c_z}(i)$ of each search query result $i$ for query $c_z$, we used its brief *textual summary*, i.e., the snippet returned by Google to describe the respective query result. These snippets typically contain the title of the result page and some 2-3 lines of text that summarizes the content's relevance with respect to the query. Second, instead of using the snippet only, we downloaded the *full document* associated with each query result $i$ for query concept $c_z$.

Next, we applied Porter-stemming [1] and stop-word removal [1] to the first 100 search results (both snippets and full document) for all 260 queries crawled from Google Suggest. Both setups, i.e., snippet- and document-based, were further subdivided into two variations each: While term frequency (TF) [1] was always applied to all index term vectors, inverse document frequency (IDF) was first switched on and then off for snippet- and document-based. Hence, we get four different setups.

## 5.4 Experiments

For comparing the performance across all metrics, we again followed the approach proposed by Resnik [15, 16] and Li *et al.* [10], i.e., we computed the predicted proximity of all concept pairs for set $B_0$ as well as $B_1$, thus obtaining the respective metric's rating vector. Next, we computed the Pearson correlation $p(\vec{v}_m, \vec{v}_i)$ of each metric $m$'s rating vector $\vec{v}_m$ with the proximity rating vectors of all participants $i \in B_z$ for one given experiment $B_z$ and averaged the summed coefficients:

$$p_z^m = \sum_{i \in B_z} p(\vec{v}_m, \vec{v}_i) \cdot \frac{1}{|B_z|} \qquad (8)$$

The correlation thus measures the metric's compliance with human ratings. The higher the average correlation, the better.[4]

Since Li *et al.* and Ziegler *et al.* demand tuning parameters, we conducted two separate runs. The first one, operating on $B_0$, was meant for parameter learning. The learned optimum parameters were then used for the second run, based on $B_1$, i.e., the actual test set.

---

[4]Opposed to [15], the inter-subject correlation does *not* represent an upper bound for metric correlation with human ratings, as can be shown easily.

### 5.4.1 Parameter Learning

Li *et al.* [10] give $\alpha = 0.2$ and $\beta = 0.6$ as optimal parameters for their approach. However, since we are supposing a different data set, we ran parameterization trials for $\alpha$ and $\beta$ again. We thereby assumed $|q^{c_z}| = 30$ for all concepts $c_z$, i.e., 30 query results were considered for defining each concept/instance $c_z$. For both $\alpha$ and $\beta$, we tested the interval $[0, 1]$ in .05 increments on $B_0$. The two-dimensional curve is shown in Figure 2. As optimal parameters we obtained $\alpha = 0.2$ and $\beta = 0.8$, which comes close to the values found by Li *et al.* [10]. The peak correlation amounts to 0.6451.

For our own approach, three parameters had to be learned, namely coefficients $\gamma$, $\delta$ and half-life $\alpha$. Again, we assumed $|q^{c_z}| = 30$. Parameters $\gamma$ and $\delta$ were determined first, having $\alpha = 10$, see Figure 3(a). The optimal values were $\gamma = 0.7$ and $\delta = 0.15$, giving the curve's peak correlation of 0.6609. As Figure 3(a) shows, all higher values are settled around some fictive diagonal. For probing half-life $\alpha$, we therefore selected two points spanning the fictive diagonal, with the optimal parameters $\gamma = 0.7$ and $\delta = 0.15$ in the middle. The results for increasing $\alpha$ over all three 2D-points are shown in Figure 3(b). Again, the peak is reached for $\gamma = 0.7$ and $\delta = 0.15$, when assuming $\alpha = 7$.

The learned values were then used in the actual evaluation runs performed on $B_1$.

### 5.4.2 Performance Analysis

First, we evaluated the proximity prediction performance across all taxonomy-based metrics. To this end, we tested all four metrics on varying query result sizes $|q(c_z)|$ for all concepts $c_z$, ranging from 1 to 80 documents. Results are displayed in Figure 4(a), giving the average correlation with human ratings for each metric and $|q^{c_z}| \in [1, 80]$. The number of topics/documents for characterizing one concept/instance appears to have little impact on Leacock-Chodorow. When $|q^{c_z}| > 40$, the correlation seems to worsen. For Li *et al.*, an increase of $|q^{c_z}|$ has merely marginal positive effects. We owe these two observations to the fact that WORDNET metrics are not designed to compare sets or lists of topics, but rather two singleton topics only (see Section 4.1). Besides, Figure 4(a) also shows that Li *et al.* has much higher correlation with human ratings than Leacock-Chodorow's simplistic metric.

For our own approach, we tested the metric's performance when using half-life $\alpha = 7$, which had been found the optimal value before, and $\alpha = \infty$. Note that an infinite half-life $\alpha = \infty$ effectively makes all topics obtain equal weight, no matter which list position they appear at. For $\alpha = 7$, the curve flattens when $|q^{c_z}| > 25$. The flattening effect appears since all topics with low ranks $i > 7$ have so little weight, less than 50% of the top position's weight. Adding more topics, considering that additional topics have increasingly worse ranks, therefore exerts marginal impact only. On the other hand, for $\alpha = \infty$, smaller fluctuations persist. This makes sense, for every added topic has *equal* impact. However, the curves for $\alpha = 7$ and $\alpha = \infty$ exhibit differences of smaller extent only. When assuming more than 40 topics per concept, correlation worsens somewhat, indicated through the $\alpha = \infty$ curve. As opposed to both WORDNET metrics, our metric performs better when offered more information, i.e., more topics per concept. The latter finding backs our design goal geared towards *multi-class* categorization (see Section 4.2).
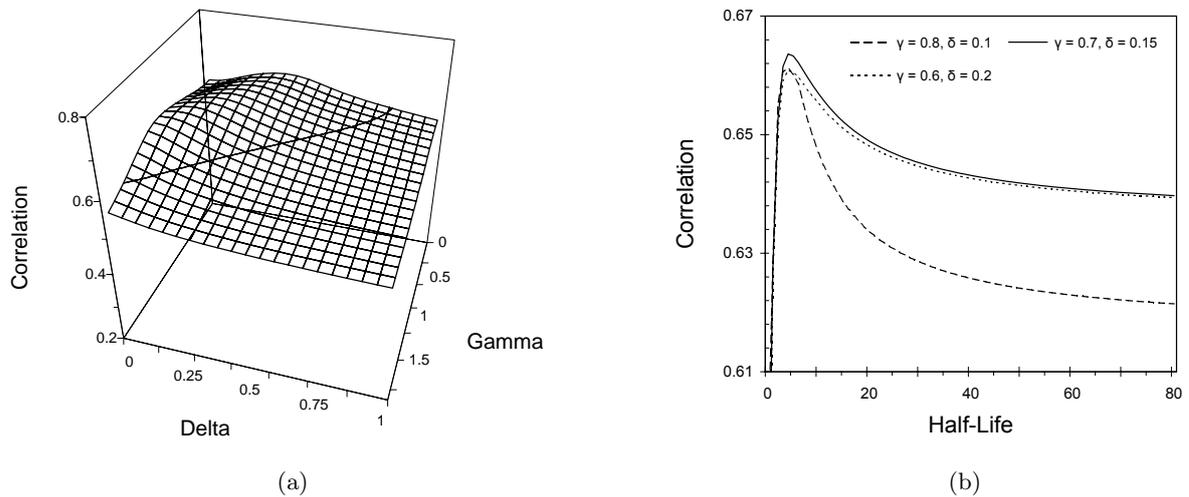
(a)



(b)

**Figure 3: Learning parameters for Ziegler *et al.***

Moreover, Figure 4(a) shows that Ziegler *et al.* performs *significantly* better than both other taxonomy-based benchmarks. With $\alpha = \infty$, the peak correlation of 0.7505 is reached for $|q^{c_z}| = 31$. Curve $\alpha = 7$ levels out around $|q^{c_z}| = 30$, giving a correlation of 0.7382. For comparison, Li *et al.*'s peak value amounts to 0.6479, Leacock-Chodorow's maximum lies at 0.5154.

Next, we compared the performance of *text*-based proximity metrics, shown in Figure 4(b). All metrics, except for full text-based with TF and IDF, drastically improve when offered more documents for representing one concept. However, fluctuations are much stronger than for the taxonomy-based metrics. For more than 20 documents per concept, snippets-based with TF and IDF performs best, reaching its maximum correlation of 0.6510 for 73 documents. This performance is comparable to Li *et al.*'s, but while the text-based metric becomes more accurate for document numbers $> 20$, the mentioned taxonomy-based metric exhibits better performance for topic numbers $< 20$.

### 5.4.3 Conclusion

We have shown that our novel metric outperforms both state-of-the-art taxonomy-based proximity metrics as well as typical text-based approaches. For reasonably large numbers of topics , i.e., $|q^{c_z}| > 20$, the correlation with human ratings lies between 0.72 and 0.75, indicating strong correlation. Leacock-Chodorow, being an utter simplistic taxonomy-based approach, and the full text-based approach with TF *and* IDF, both exhibited correlations below 0.5 for more than 20 topics/documents. Opposed to our approach, their performance was better when using *less* information, i.e., $< 20$ topics/documents, still peaking only slightly above 0.5. The other three metrics, i.e., Li *et. al.*, snippet-based with and without IDF, and full text-based without IDF, had correlation scores between 0.55 and 0.65 for more than 20 topics/documents.

## 6. OUTLOOK AND FUTURE WORK

Semantic proximity metrics are becoming an increasingly important component for frameworks geared towards the machine's understanding of human-created sources of information, such as the Web. Currently, only for small portions of information fragments, namely words and simple concepts stored in thesauri and dictionaries such as WordNet, semantic similarity measures are applicable. By harnessing the combined power of both Google and ODP, we were able to extend semantic proximity to *arbitrary* concepts, e.g., names of persons, composed concepts, song titles, and so forth. Moreover, we introduced a new taxonomy-based proximity metric that showed significantly better performance than existing state-of-the-art approaches and comes close to human judgement.

For the future, we would like to steer our research towards the nature of proximity itself. In other words, besides revealing that two concepts are somewhat related, we would like to reveal the *type* of their mutual relationship. For instance, when supposing the concepts SPACE SHUTTLE and CAPE CANAVERAL, an algorithm should inform us that both are related because Space Shuttles are LOCATED IN Cape Canaveral, and that they are both from the NASA universe.

## Acknowledgements

## 7. REFERENCES

[1] BAEZA-YATES, R., AND RIBEIRO-NETO, B. *Modern Information Retrieval.* Addison-Wesley, Reading, MA, USA, May 1999.

[2] BREESE, J., HECKERMAN, D., AND KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence* (Madison, WI, USA, July 1998), Morgan Kaufmann, pp. 43–52.
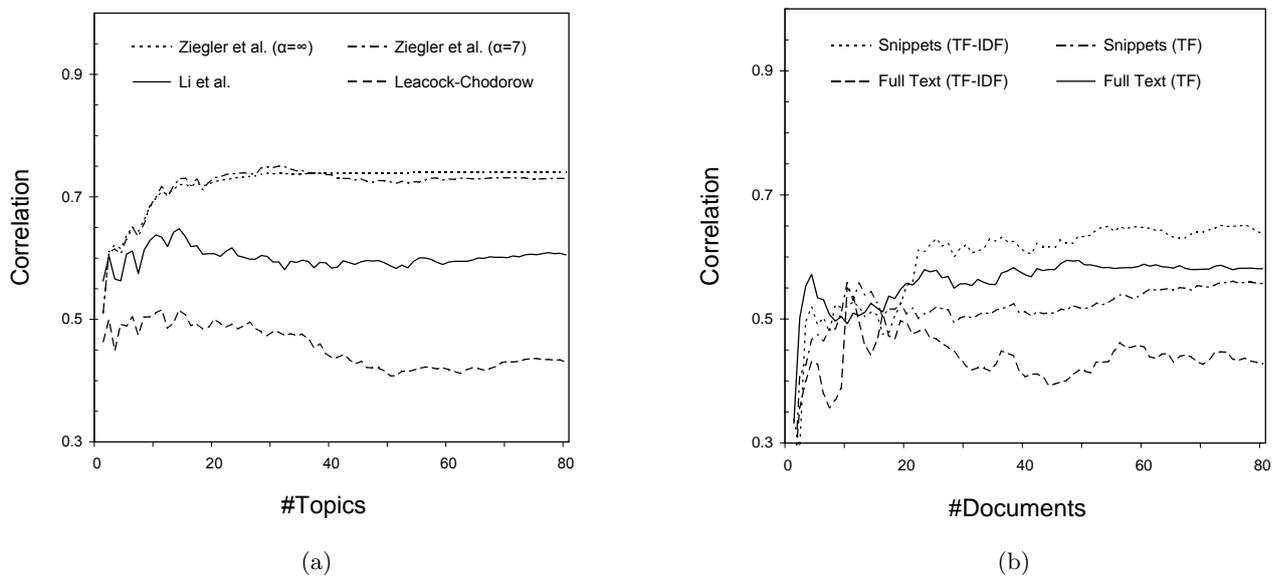
Figure 4: Correlations of metrics with human ratings

[3] BUDANITSKY, A., AND HIRST, G. Semantic distance in Word-Net: An experimental, application-oriented evaluation of five measures. In *Proceedings of the Workshop on WordNet and Other Lexical Resources* (Pittsburgh, PA, USA, June 2000).

[4] CHIEN, S., AND IMMORLICA, N. Semantic similarity between search engine queries using temporal correlation. In *Proceedings of the 14th International World Wide Web Conference* (Chiba, Japan, May 2005), ACM Press.

[5] CHIRITA, P.-A., NEJDL, W., PAIU, R., AND KOHLSCHÜTTER, C. Using odp metadata to personlize search. In *Proceedings of the 28th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Salvador, Brazil, August 2005), ACM Press.

[6] CIMIANO, P., HANDSCHUH, S., AND STAAB, S. Towards the self-annotating web. In *Proceedings of the 13th International World Wide Web Conference* (New York, NY, USA, 2004), ACM Press, pp. 462–471.

[7] CIMIANO, P., LADWIG, G., AND STAAB, S. Gimme' the context: context-driven automatic semantic annotation with c-pankow. In *Proceedings of the 14th International World Wide Web Conference* (Chiba, Japan, 2005), ACM Press, pp. 332–341.

[8] GANESAN, P., GARCIA-MOLINA, H., AND WIDOM, J. Exploiting hierarchical domain structure to compute similarity. *ACM Transactions on Information Systems 21*, 1 (2003), 64–93.

[9] JIANG, J., AND CONRATH, D. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics* (Taiwan, 1997).

[10] LI, Y., BANDAR, Z., AND MCLEAN, D. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering 15*, 4 (2003), 871–882.

[11] LIN, D. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning* (San Francisco, CA, USA, 1998), Morgan Kaufmann, pp. 296–304.

[12] MAGUITMAN, A., MENCZER, F., ROINESTAD, H., AND VESPIGNANI, A. Algorithmic detection of semantic similarity. In *Proceedings of the 14th International World Wide Web Conference* (Chiba, Japan, 2005), ACM Press, pp. 107–116.

[13] MILLER, G. Wordnet: A lexical database for english. *Communications of the ACM 38*, 11 (1995), 39–41.

[14] MILLER, G., AND CHARLES, W. Contextual correlates of semantic similarity. *Language and Cognitive Processes 6*, 1 (February 1991), 1–28.

[15] RESNIK, P. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (Montreal, Canada, 1995), pp. 448–453.

[16] RESNIK, P. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research 11* (1999), 95–130.

[17] RUBENSTEIN, H., AND GOODENOUGH, J. Contextual correlates of synonymy. *Communications of the ACM 8*, 10 (1965), 627–633.

[18] SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference* (Hong Kong, China, May 2001).

[19] VAN RIJSBERGEN, K. *Information Retrieval*. Butterworths, London, UK, 1975.

[20] VLACHOS, M., MEEK, C., VAGENA, Z., AND GUNOPULOS, D. Identifying similarities, periodicities and bursts for online search queries. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data* (Paris, France, 2004), ACM Press, pp. 131–142.

[21] WEN, J.-R., NIE, J.-Y., AND ZHANG, H.-J. Clustering user queries of a search engine. In *Proceedings of the 10th International World Wide Web Conference* (Hong Kong, China, 2001), ACM Press, pp. 162–168.

[22] ZIEGLER, C.-N., LAUSEN, G., AND SCHMIDT-THIEME, L. Taxonomy-driven computation of product recommendations. In *Proceedings of the 2004 ACM CIKM Conference on Information and Knowledge Management* (Washington, D.C., USA, November 2004), ACM Press, pp. 406–415.

[23] ZIEGLER, C.-N., MCNEE, S., KONSTAN, J., AND LAUSEN, G. Improving recommendation lists through topic diversification. In *Proceedings of the 14th International World Wide Web Conference* (Chiba, Japan, May 2005), ACM Press.

| Word Pair | | ∅-Rating | Std. Dev. |
|---|---|---|---|
| EMINEM | GREEN DAY | 3.304 | 0.997 |
| XML | VODAFONE | 1.348 | 0.476 |
| KING | JOBS | 1.522 | 0.714 |
| HOROSCOPES | QUOTES | 1.826 | 0.761 |
| FIREFOX | INTERNET EXPLORER | 4.391 | 0.57 |
| IMDB | BLOCKBUSTER | 3.304 | 0.856 |
| DOGS | DISNEY | 2.478 | 0.926 |
| FLOWERS | FOOD NETWORK | 1.609 | 0.82 |
| E-CARDS | VALENTINE'S DAY | 3.435 | 0.825 |
| FREE MUSIC DOWNLOADS | ITUNES | 3.696 | 0.906 |
| DELTA AIRLINES | US AIRWAYS | 4.0 | 0.978 |
| DELL | BEST BUY | 3.13 | 0.947 |
| VODAFONE | O2 | 4.304 | 0.687 |
| MOVIES | NEWS | 2.435 | 1.056 |
| YAHOO MAPS | ZONE ALARM | 1.696 | 0.953 |
| DICTIONARY | SUPER BOWL | 1.13 | 0.448 |
| POEMS | LYRICS | 4.087 | 0.83 |
| QUICKEN | PAYPAL | 2.609 | 0.872 |
| NASA | KAZAA LITE | 1.043 | 0.204 |
| LAS VEGAS | EXCHANGE RATES | 1.913 | 0.88 |
| CARS | GIRLS | 2.043 | 1.083 |
| WEATHER CHANNEL | NEWS | 3.435 | 0.77 |
| GUITAR TABS | HOTMAIL | 1.13 | 0.448 |
| QUIZ | AMAZON | 1.217 | 0.507 |
| TSUNAMI | WEATHER | 3.87 | 0.74 |

**Table 1: Training set $B_0$, along with human rating averages and standard deviation**

| Word Pair | | ∅-Rating | Std. Dev. |
|---|---|---|---|
| HOLIDAY INN | VALENTINE'S DAY | 1.882 | 0.855 |
| BLOCKBUSTER | NIKE | 1.588 | 0.867 |
| INLAND REVENUE | LOVE | 1.216 | 0.604 |
| GOOGLE | GMAIL | 4.118 | 0.783 |
| LOVE QUOTES | TV GUIDE | 1.549 | 0.749 |
| PC WORLD | UNITED AIRLINES | 1.235 | 0.468 |
| JOKES | QUOTES | 2.294 | 1.108 |
| DELTA AIRLINES | LOVE POEMS | 1.098 | 0.357 |
| BRITNEY SPEARS | PARIS HILTON | 3.804 | 0.767 |
| NASA | SUPER BOWL | 1.392 | 0.659 |
| PERIODIC TABLE | TOYOTA | 1.176 | 0.55 |
| WINZIP | ZIP CODE FINDER | 1.902 | 1.241 |
| EASYJET | CHEAP FLIGHTS | 4.294 | 0.749 |
| PEOPLE SEARCH | WHITE PAGES | 3.843 | 1.274 |
| TSUNAMI | HARRY POTTER | 1.098 | 0.297 |
| BBC | JENNIFER LOPEZ | 1.686 | 0.779 |
| U2 | RECIPES | 1.157 | 0.364 |
| THESAURUS | US AIRWAYS | 1.118 | 0.322 |
| XBOX CHEATS | YAHOO GAMES | 2.941 | 0.998 |
| CURRENCY CONVERTER | EXCHANGE RATES | 4.137 | 0.971 |
| FLOWERS | WEATHER | 2.765 | 1.059 |
| USED CARS | VIRGIN | 1.431 | 0.693 |
| EMINEM | MUSIC | 4.137 | 0.687 |
| CARS | HONDA | 4.176 | 0.617 |
| LYRICS | REAL PLAYER | 2.588 | 1.07 |
| FREE GAMES | XBOX 2 | 2.549 | 0.996 |
| MSN MESSENGER | UPS | 1.706 | 0.799 |
| MICROSOFT | INTERNET EXPLORER | 4.314 | 0.727 |
| POEMS | SONG LYRICS | 3.647 | 0.762 |
| DELTA AIRLINES | WALMART | 1.725 | 0.794 |

Table 2: Test set $B_1$, along with human rating averages and standard deviation