Finding Similar Documents in Document Collections

Thorsten Brants and Reinhard Stolle

Palo Alto Research Center (PARC)
3333 Coyote Hill Rd, Palo Alto, CA 94304, USA
{brants,stolle}@parc.com

Abstract

Finding similar documents in natural language document collections is a difficult task that requires general and domain-specific world knowledge, deep analysis of the documents, and inference. However, a large portion of the pairs of similar documents can be identified by simpler, purely word-based methods. We show the use of Probabilistic Latent Semantic Analysis for finding similar documents. We evaluate our system on a collection of photocopier repair tips. Among the 100 top-ranked pairs, 88 are true positives. A manual analysis of the 12 false positives suggests the use of more semantic information in the retrieval model.

1. Introduction

Collections of natural language documents that are focused on a particular subject domain are commonly used by communities of practice in order to capture and share knowledge. Examples of such "focused document collections" are FAQs, bug-report repositories, and lessons-learned systems. As such systems become larger and larger, their authors, users and maintainers increasingly need tools to perform their tasks, such as browsing, searching, manipulating, analyzing and managing the collection. In particular, the document collections become unwieldy and ultimately unusable if obsolete and redundant content is not continually identified and removed.

We are working with such a knowledge-sharing system, focused on the repair of photocopiers. It now contains about 40,000 technician-authored free text documents, in the form of tips on issues not covered in the official manuals. Such systems usually support a number of tasks that help maintain the utility and quality of the document collection. Simple tools, such as keyword search, for example, can be extremely useful. Eventually, however, we would like to provide a suite of tools that support a variety of tasks, ranging from simple keyword search to more elaborate tasks such as the identification of "duplicates." Fig. 1 shows a pair of similar tips from our corpus. These two tips are about the same problem, and they give a similar analysis as to why the problem occurs. However, they suggest different solutions: Tip 118 is the "official" solution, whereas Tip 57 suggests a short-term "work-around" fix to the problem. This example illustrates that "similarity" is a complicated notion that cannot always be measured along a one-dimensional scale. Whether two or more documents should be considered "redundant" critically depends on the task at hand. In the example of Fig. 1, the work-around tip may seem redundant and obsolete to a technician who has the official new safety cable available. In the absence of this official part, however, the work-around tip may be a crucial piece of information.

Our goal is to develop techniques that analyze the conceptual contents of natural language documents at a granularity that is fine enough to capture distinctions like the one between Tips 57 and 118, described in the previous paragraph. In order to do that, we are designing formal representations of document contents that will allow us to assess not only whether two documents are about the same subject but also whether two documents actually say the same thing. We are currently focusing on the tasks of computer-assisted redundancy resolution. We hope that our techniques will eventually extend to support even more ambitious tasks such as the identification and resolution of inconsistent knowledge, knowledge fusion, question answering, and trend analysis.

We believe that, in general, the automated or computer-assisted management of collections of natural language documents requires a fine-grained analysis and representation of the documents' contents. This fine granularity in turn mandates deep linguistic processing of the text and inference capabilities using extensive linguistic and world knowledge. Following this approach, our larger research group has implemented a prototype, which we will briefly describe in the next section. This research prototype system is far from complete. Meanwhile, we are investigating to what extent currently operational techniques are useful to support at least some of the tasks that arise from the maintenance of focused document collections. We have investigated the utility of Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999b) for the task of finding similar documents. Section 3. describes our PLSA model and Section 4. reports on our experimental results in the context of our corpus of repair tips. In that section, we also attempt to characterize the types of similarities that are easily detected and contrast them to the types that are easily missed by the PLSA technique. Finally, we speculate how symbolic knowledge representation and inference techniques that rely on a deep linguistic analysis of the documents may be coupled with statistical techniques in order to improve the results.

2. Knowledge-Based Approach

Our goal is to build a system that supports a wide range of knowledge management tasks for focused document collections. We believe that powerful tools for tasks like redundancy resolution, topic browsing, question answering, knowledge fusion, and so on, need to analyze and represent the documents' conceptual contents at a fine level of granularity.

Concentrating on the task of redundancy resolution, our

Tip 57		Tip 118	
Problem:	Left cover damage	Problem:	The current safety cable used in the 5100 Document Handler fails prematurely, causing the Left Document Handler Cover to break.
Cause:	The left cover safety cable is breaking, allowing the left cover to pivot too far, breaking the cover.	Cause:	The plastic jacket made the cable too stiff. This causes stress to be concentrated on the cable ends, where it eventually snaps.
Solution:	Remove the plastic sleeve from around the cable. Cutting the plastic off of the cable makes the cable more flexible, which prevents cable breakage. Cable breakage is a major source of damage to the left cover.	Solution:	When the old safety cable fails, replace it with the new one, which has the plastic jacket shortened.

Figure 1: Example of Eureka tips

project group has so far built a prototype whose goal is to identify conceptually similar documents, regardless of how they are written. This task requires extensive knowledge about language and of the world. Since most of this knowledge engineering effort is performed by hand at the moment, our system's coverage is currently limited to fifteen pairs of similar tips. We are in the process of scaling the system up by one to two orders of magnitude. Eventually, we hope to also support more general tasks, namely identify the parts of two documents that overlap; and identify parts of the documents that stand in some relation to each other, such as expanding on a particular topic or being in mutual contradiction. Such a system will enable the maintenance of vast document collections by identifying potential redundancies or inconsistencies for human attention.

State-of-the-art question answering and information extraction techniques (e.g., (Bear et al., 1997)) are sometimes able to identify entities and the relations between them at a fine level of granularity. However, the functionality and coverage of these techniques is typically restricted to a limited set of types of entities and relations that have been formalized upfront using static templates. Like a small number of other research projects (e.g., the TACITUS project (Hobbs et al., 1993)), our approach is based on the belief that the key to solving this problem is a principled technique for producing formal representations of the conceptual contents of the natural language documents. In our approach, a deep analysis based on Lexical Functional Grammar theory (Kaplan and Bresnan, 1982) combined with Glue Semantics (Dalrymple, 1999) produces a compact representation of the syntactic and semantic structures for each sentence. From this language-driven representation of the text, we map to a knowledge-driven representation of the contents that abstracts away from the particular natural language expression. This mapping includes several—not necessarily sequential—steps. In one step, we rely on a domain-specific ontology to identify canonicalized entities and events that are talked about in the text. In our case, these entities and events include things like parts, e.g., photoreceptor belt, and relevant activities such as cleaning, for example. Another step performs thematic role assignments and assembles fragments of conceptual structures from the normalized entities and events (e.g., cleaning a photoreceptor belt). Furthermore, certain relations are normalized; for example, "stiff" and "flexible" (in Fig. 1) both refer to the rigidity of an object, one being the inverse of the other. Yet another step composes structure fragments into higher-level structures that reflect causal or temporal relations, such as action sequences or repair plans. All steps involve ambiguity resolution as a central problem, which requires inference based on extensive linguistic and world knowledge. For a more detailed description of this approach and its scalability, see (Crouch et al., 2002).

Finally, we assess the similarity of two documents using a variant of the Structure Mapping Engine (SME) (Forbus et al., 1989). SME anchors its matching process in identical elements that occur in the same structural positions in the base and target representations, and from this builds a correspondence. The larger the structure that can be recursively constructed in this manner, while preserving a systematicity constraint of one-to-one correspondence between base and target elements and the identicality of anchors, the greater the similarity score.

We expect that the fine-grained conceptual representations discussed in this section will eventually enable our system to detect whether two documents are not only about the same subject but also saying the same thing. Many interesting cases of similarity can, however, be detected with lighter-weight techniques. This is the topic of the next section.

3. The Word-Based Statistical Model

While in the general case deep processing, knowledge about the world, and inference are necessary to identify similar documents, there may be a large number of similar pair that can be discovered by a shallow approach. We now view the task of finding similar pairs of documents as an information retrieval problem where documents are matched based on the words that occur in the documents, i.e., we use a vector space model of the documents. Comparison is done using Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999b).

3.1. Document Preprocessing

Each document is first preprocessed by:

1. Separating the document fields. Each tip usually comes with additional administrative information like author, submission date, location, status, contact information, etc. We extract the information that is contained in the `CHAINS`, `PROBLEM`, `CAUSE`, and `SO-`

LUTION fields[1].

2. Tokenizing the document. Words and numbers are separated at white space, punctuation is stripped, abbreviations are recognized.

3. Lemmatizing each token, i.e., each word is uniquely mapped to a base form. We use the LinguistX lemmatizer[2] to perform this task.

Steps 1 to 3 identify the terms in the vocabulary. We select the subset of those terms that occur in at least two documents. Given this vocabulary, each document d is represented by its term-frequency vector $f(d,w)$, where w are the terms of the document.

3.2. Probabilistic Latent Semantic Indexing

Probabilistic Latent Semantic Analysis (PLSA) is a statistical latent class model or aspect model (Hofmann, 1999a; Hofmann, 1999b). It can be seen as a statistical view of Latent Semantic Analysis (LSA) (Deerwester et al., 1990). The model is fitted to a training corpus by the Expectation Maximization (EM) algorithm (Dempster et al., 1977). It assigns probability distributions over classes to words and documents and thereby allows them to belong to more than one class, and not to only one class as is true of most other classification methods. PLSA represents the joint probability of a document d and a word w based on a latent class variable z:[3]

$$P(d,w) = P(d)\sum_z P(w|z)P(z|d) \tag{1}$$

The model makes an independence assumption between word w and document d if the latent class z is given, i.e., $P(w|z,d) = P(w|z)$. PLSA has the following view of how a document is generated: first a document $d \in \mathcal{D}$ (i.e., its dummy label) is chosen with probability $P(d)$. For each word in document d, a latent topic $z \in \mathcal{Z}$ is chosen with probability $P(z|d)$, which in turn is used to choose the word $w \in \mathcal{W}$ with probability $P(w|z)$.

A model is fitted to a document collection $\mathcal{D}$ by maximizing the log-likelihood function $\mathcal{L}$:

$$\mathcal{L} = \sum_{d\in\mathcal{D}}\sum_{w\in d} f(d,w) log P(d,w) \tag{2}$$

The E-step in the EM-algorithm is

$$P(z|d,w) = \frac{P(z)P(d|z)P(w|z)}{\sum_{z'} P(z')P(d|z')P(w|z')} \tag{3}$$

and the M-step consists of

$$P(w|z) = \frac{\sum_d f(d,w)P(z|d,w)}{\sum_{d,w'} f(d,w')P(z|d,w')} \tag{4}$$

$$P(d|z) = \frac{\sum_w f(d,w)P(z|d,w)}{\sum_{d',w} f(d',w)P(z|d',w)} \tag{5}$$

$$P(z) = \frac{\sum_{d,w} f(d,w)P(z|d,w)}{\sum_{d,w} f(d,w)} \tag{6}$$

The parameters are either randomly initialized or according to some prior knowledge.

After having calculated the reduced dimensional representations of documents in the collection, we map the vectors back to the original term space to yield vectors $P(w|d)$ by

$$P(w|d) = \sum_z P(w|z)P(z|d) \tag{7}$$

$P(w|d)$ can be seen as a smoothed version of the empirical distribution $r(w|d) = f(d,w)/f(d)$ of words in the document. The advantage of the smoothed version is that it captures semantic similarities through the lower-dimensional representation.

Note that this process is intended for the pairwise comparison of all documents in the training collection. It can be extended to new documents q (query or test documents) by using the folding-in process. Folding-in uses Expectation-Maximization as in the training process; the E-step is identical, the M-step keeps all the $P(w|z)$ constant and recalculates $P_{fi}(z|q)$. Usually, a very small number of iterations is sufficient for folding-in. We get a smoothed representation of a folded-in document by

$$P_{fi}(w|q) = \sum_z P(w|z)P_{fi}(z|q) \tag{8}$$

This corresponds to the PLSI-U model described in (Hofmann, 1999b).

3.3. Document Comparison

A standard way of comparing vector space representations of documents d_1 and d_2 is to calculate the cosine similarity score of tf-idf weighted document vectors (Salton, 1988):

$$\mathrm{sim}_{\cos}(d_1,d_2) = \frac{\sum_w \hat{f}(d_1,w)\hat{f}(d_2,w)}{\sqrt{\sum_w \hat{f}(d_1,w)^2}\sqrt{\sum_w \hat{f}(d_2,w)^2}} \tag{9}$$

$\hat{f}(d,w)$ is the weighted frequency of word w in document d:

$$\hat{f}(d,w) = f(d,w)\log\frac{N}{df(w)} \tag{10}$$

where N is the total number of documents, and $df(w)$ is the number of documents containing word w.

We additionally perform the comparison on the PLSA representation of $P(w|d)$. Pairwise comparisons are done by

$$\mathrm{sim}^{\mathrm{PLSA}}_{\cos}(d_1,d_2) = \frac{\sum_w P(w|d_1)P(w|d_2)}{\sqrt{\sum_w P(w|d_1)^2}\sqrt{\sum_w P(w|d_2)^2}} \tag{11}$$

[1]The CHAINS field contains a numerical identifier of the product line.

[2]For information about the LinguistX tools, see www.inxight.com/products/linguistx/

[3]Unless otherwise noted, we use the following notational conventions: training documents $d, d' \in \mathcal{D}$, test documents $q, q' \in \mathcal{Q}$, words $w, w' \in \mathcal{W}$, and classes $z, z' \in \mathcal{Z}$.

Table 1: Precision of the statistical model for the n top-ranked pairs. A pair of tips is considered a "true positive" if their conceptual contents are categorized to be the same, similar, or in the subset relationship.

n	precision
10	100%
20	100%
30	100%
40	96%
50	92%
60	92%
70	90%
80	87%
90	88%
100	88%

Both similarities are combined with a weight λ to yield the final similarity score (see (Hofmann, 1999b)).

$$\mathrm{sim}(d_1, d_2) = \lambda \mathrm{sim}_{\cos}(d_1, d_2) + (1 - \lambda)\mathrm{sim}_{\cos}^{\mathrm{PLSA}}(d_1, d_2) \quad (12)$$

The output of the algorithm is a list of pairs ranked according to their similarity.

4. Experiments

We applied the algorithm described in Section 3. to a subset of the Eureka database consisting of 1,321 tips. PLSA representations of $P(w|d)$ were created for each tip, and pairs of tips were ranked according to their similarity. Following (Hofmann, 1999b), we created models with $Z = 32, 48, 64, 80, 128$ latent classes, calculated the average $P(w|d)$. The similarity score was combined with the standard tf-idf cosine similarity with a weight of $\lambda = \frac{1}{6}$.

4.1. Precision and Recall

We manually inspected the 100 top-ranked pairs of tips and classified their similarity by hand according to the types of similarity described in Section 4.2.. The results are shown in Table 1. Of the 10 top-ranked pairs, all 10 were actual duplicates,[4] of the 40 top-ranked pairs, 96% were true positives, and so on. The manual inspection of the 100 top-ranked pairs (of the potential 871,860 pairs) revealed 88 true positives.

Independent manual sampling of the subset of 1,321 tips, which is a very tedious and time-consuming task, revealed 17 similar pairs (14 pairs and 1 triple). 3 of these pairs were among the top 100 emitted by the word-based statistical model. This is a recall of 18% on the manually identified similar pairs. However, it is unclear how this number relates to the overall recall because the distribution of the other similar pairs is currently unclear.

[4] A pair of tips is considered "duplicates" if their conceptual contents are categorized to be the same. A pair of tips is considered a "true positive" if their conceptual contents are categorized to be the same, similar, or in the subset relationship. See Section 4.2..

Table 2: Number of pairs with structural and conceptual match in the 100 top-ranked pairs of documents. We are interested in finding the conceptually same/similar/subset pairs. False positives are shown in *italics*.

		conceptual				
		same	sim	subset	diff	sum
surface	same	24	0	10	*2*	36
	sim	17	24	13	*8*	62
	diff	0	0	0	*2*	2
	sum	41	24	23	*12*	100

4.2. Types of Similarity

The word-based statistical model of Section 3. seems to be good at identifying pairs whose texts are similar *at a surface level*. In order to see how well the model does at identifying pairs whose contents are *conceptually similar*, we manually performed a qualitative evaluation and classified each of the 100 top-ranked pairs according to the following criteria:

Surface similarity of texts: *same, similar, different*. Surface similarity describes the similarity of the set of words and syntactic constructions used in the documents. *Same* means that the documents are (almost) identical. *Similar* means that some words may be different or replaced by synonyms (e.g., "fault" vs. "failure" vs. "problem", "motor" vs. "drive", "line" vs. "wire", etc.), constructions are different, order of sentences may be different. *Different* means that the texts are different.

Conceptual similarity of contents: *same, similar, subset, different*. Conceptual similarity refers to the semantic/conceptual contents of the document, independent of how it is expressed as surface text. *Same* means that the documents have (almost) the same contents (e.g., "cutting the plastic off of the cable makes the cable more flexible" vs. "the plastic jacket made the cable too stiff"). *Similar* means that there is a significant overlap of conceptual contents between the two documents; for example, the tips describe the same problem but suggest different solutions (see Fig. 1), or, the tips describe an analogous problem exhibited at different mechanical parts (see Fig. 2).

Subset describes cases where the conceptual contents of one document form a proper subset of the conceptual contents of the other document—for example, if one document elaborates on the other. *Different* describes conceptually different documents.

Table 2 shows how many of the pairs fall into the different categories. Since the PSLA model is word-based, almost none of the pairs have different surface similarity. In the 100 top-ranked pairs, the majority of false positives occur when the surface texts are similar but the conceptual contents are different (8 out of 12).

The algorithm identifies surface similarity very well, only 2 out of 100 pairs are different at the surface text level.

Tip 690		Tip 714	
Problem:	08-110, Tray 3 misfeed	Problem:	08-100, Tray 1 misfeed
Cause:	J201 Pin 1 loose. Drive coupling set screw loose, Blower hose came off, Fang plate out of adjustment, Stack height out of adjustment, Defective DRCC1.	Cause:	Set screw on feed clutch loose. Stack height sensor out of bracket. Feeder drive coupling loose. Blower hose off.
Solution:	Reseat J201 Pin 1. Tighten drive coupling, Reconnect blower hose , Adjust fang plate, Adjust stack height. Replace DRCC1.	Solution:	Adjust clutch. Repair stack height sensor. Tighten feeder driv e coupling. Repair blower hose.

Figure 2: True positive: this pair at rank 68 has similar surface text and is similar at the conceptual level.

Tip 1280		Tip 1281	
Problem:	Xerox Binder 120. The "READY FOR AUTO FEED" message does not change when set clamp assy is pulled in	Problem:	Xerox Binder 120. The Binder 120 does not display "Ready for auto feed" message.
Cause:	Set Clamp extended sensor (Q23) is "H" all the time	Cause:	Set Clamp extended sensor (Q23) is "Lo" all the time
Solution:	check the set clamp sensor wires for an open circuit, if ok, Replace the set clamp extended sensor (Q23)	Solution:	Check the set clamp extended sensor wires for Short circuit to frame, Set clamp out flag is in the sensor correctly, if ok, replace the sensor.

Figure 3: False positive: this pair at rank 37 has almost the same surface text but is different at the conceptual level.

These two pairs involve very long documents (average of 1030 tokens per document compared to 132 tokens per document overall average). The documents have an overlap in vocabulary, but the sentences and sequences of sentences are very different.

Correlation with conceptual similarity can also be found, but it is smaller. 10 out of 100 pairs were categorized as the same or similar at the surface but are conceptually different; from the viewpoint of a user in the context of a conceptual task, these pairs should not be identified as similar tips. We believe that a deeper analysis of the document contents as outlined in Section 2. will help distinguish between conceptually different documents and, therefore, reduce the number of such false positives.

One of the two pairs that are almost the same at the surface level but have different conceptual contents is shown in Fig. 3.

They use the same or very similar words, but make opposite statements at the conceptual level. Tip 1280 describes a sensor signal that is erroneously "high" because of an open circuit. Tip 1281 describes a sensor signal that is erroneously "low" because of a short circuit. This difference cannot be found by the word-based statistical model. The topics of these two documents are very similar; however, a correct analysis of the contents requires the recognition of the difference between "does not display" and "does not change", the difference between "Lo" and "H", and the difference between "open circuit" and "short circut" despite the fact that these phrases often occur in similar contexts.

Fig. 4 shows a pair with similar surface texts but different conceptual contents. Tip 227 explains how to repair or prevent a particular failure that is caused by a ring's wearing out. Tip 173 says that an improved repair kit can be ordered; it also provides a work-around for the case in which that improved kit is not available.

The two examples in Figures 3 and 4 show that in many cases it is necessary to process the text more deeply than at the word level in order to be able to recognize fine-grained distinctions in the documents' contents. On the other hand, a large number of true positives are actually discovered by the word-based model (88 out of the 100 top-ranked pairs). The word-based statistical model even finds cases in which the conceptual contents are similar, but where this fact is not immediately obvious from the surface-level texts. Fig. 2 shows an example of this case. The two tips describe almost the same fault situation, except that one of them occurs in connection with Tray 1 while the other one occurs in connection with Tray 3. Even for a human—at least for an untrained human—, this pair is difficult to detect.

The examples suggests that symbolic and statistical techniques may be good at different tasks that complement each other nicely. Statistical techniques seem to be good at identifiying that the two tips are about the same topic. Knowledge-based techniques—specifically, a domain ontology—may help distinguish "Fuser Couplings" from the "Fuser Couplings and Shaft Repair Kit" (cf. Fig. 4), which in turn may trigger further distinctions between the two tips based on domain-specific knowledge. Similarly, the example in Fig. 3 suggests that a statistical analysis coupled with a limited normalization of relations that occur frequently in the domain may be a promising direction to pursue.

Fig. 5 shows the rank of a pair vs. its similarity. Our data set contains 1,321 documents, i.e., there are 871,860 pairs. Word-based similarity does not decrease linearly. There is a large drop at the beginning, then the curve is relatively flat, and it suddenly drops again at the very end. All of the manually found similar pairs (the 17 pairs described in Section 4.1.) are marked with a $\circ$ in the graph; they are among the first 7% (the lowest rank is 57,014). We do currently not

	Tip 173		Tip 227
Problem:	Improved Fuser Couplings 600K31031 Tag P-184. Broken calls when servicing failed Fuser Drive Couplings.	Problem:	Fuser Couplings and Shaft Repair Kit, 605K3950, Tag P-129. The retaining ring that holds the Fuser Assembly Drive Coupling in place wears out and falls off the shaft.
Cause:	The parts needed to repair a Fuser Drive failure are presently contained in two separate Kits. If the service representative does not have both Kits in inventory the service call is interrupted.	Cause:	The Fuser Assembly Drive Coupling rubs against the retaining ring as it turns.
Solution:	1. To repair Fuser Drive failures, order the new Fuser Couplings an d Shaft Repair Kit 600K31031, TAG P-184. This kit contains all the parts in Fuser Couplings and Shaft Repair Kit 605K3950 except that the improved Drive Coupling, issued separately in Kit 600K31030, has been substituted. 2. If you have 600K31030 as well as 605K3950 in inventory, these Kits can be salvaged to provide the same parts as the new Kit. Open 605K3950 and discard only the Fuser Drive Coupling, then use the Coupling contained in Kit number 600K31030 in its place.	Solution:	On the next service call check to see if P-129 is installed. If Tag P-129 is not installed, order and install the Fuser Couplings and Shaft Repair Kit, 605K3950.

Figure 4: False positive: this pair at rank 86 has similar surface text and is about similar parts, but is different at the conceptual level.

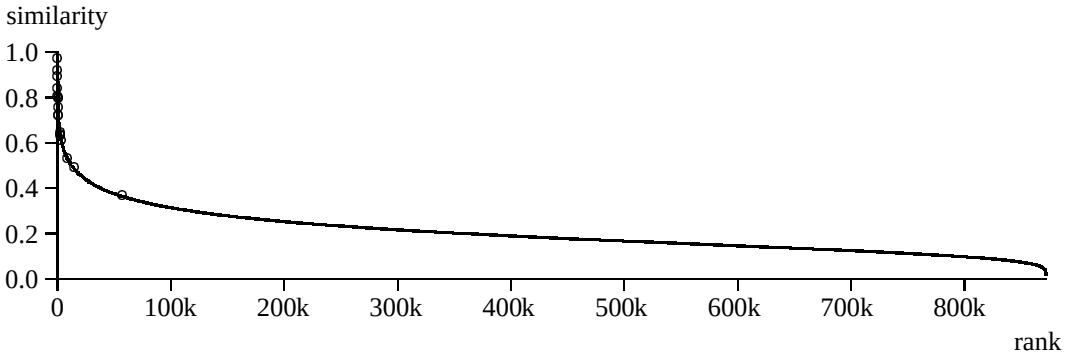

Figure 5: Rank vs. PLSA similarity. Manually found pairs are marked with ∘.

know whether there are any similar pairs below this rank, but it is probably safe to assume that almost all of the similar pairs are within the initial portion of the graph. Even if the presented statistical method does not rank all similar pairs at the very top, it seems to efficiently place them in a small initial segment at the top.

One focus of our current research effort is to understand the capabilities and limitations of the current PLSA model in order to design an improved system by, for example, (1) supplying the PLSA model with better-suited information for any given particular task, or (2) using the current version of the PLSA model as a prefilter for the knowledge-based approach.

5. Conclusions

We address the problem of matching the conceptual contents of documents. The domain of the documents in our experiments is the repair of photocopiers. In general, the problem requires world knowledge and deep processing of the documents. But in a large number of cases, similar documents can be found by shallow processing and a word-based statistical model. A quantitative evaluation shows that 88 of the 100 statistically top-ranked documents are true positives. An analysis of the erroneous cases indicates where the statistical model could benefit from deeper processing. Two important types of information that are currently absent from our statistical model are negation and relations between entities. We expect that providing the model with more semantic information along these lines will improve our system's performance and allow it to make finer distinctions among the documents' contents.

6. References

J. Bear, D. Israel, J. Petit, and D. Martin. 1997. Using information extraction to improve document retrieval. In E. M. Voorhees and D. K. Harman, editors, *The Sixth Text REtrieval Conference (TREC-6)*, pages 367–377. NIST.

R. Crouch, C. Condoravdi, R. Stolle, T. King, V. de Paiva, J. O. Everett, and D. G. Bobrow. 2002. Scalability of redundancy detection in focused document collections. In *Proceedings First International Workshop on Scalable Natural Language Understanding (SCANALU-2002)*, Heidelberg, Germany.

M. Dalrymple, editor. 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach.* MIT Press, Cambridge, MA.

S. Deerwester, S. Dumais, G. W. Furnas, T. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–21.

K. D. Forbus, B. Falkenhainer, and D. Gentner. 1989. The structure mapping engine: Algorithm and examples. *Artificial Intelligence*, 41(1):1–63.

J. R. Hobbs, M. Stickel, S. Appelt, and P. Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63(1–2):69–142.

T. Hofmann. 1999a. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, Stockholm, Sweden.

T. Hofmann. 1999b. Probabilistic latent semantic indexing. In *Proceedings of SIGIR-99*, pages 35–44, Berkeley, CA.

R. M. Kaplan and J. Bresnan. 1982. Lexical-functional grammar: A formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, MA.

G. Salton. 1988. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison Wesley.