Discovery of Novel Term Associations in a Document Collection

Teemu Hynönen, Sébastien Mahler, and Hannu Toivonen

Department of Computer Science and HIIT, University of Helsinki, Finland
firstname.lastname@cs.helsinki.fi

Abstract. We propose a method to mine novel, document-specific associations between terms in a collection of unstructured documents. We believe that documents are often best described by the relationships they establish. This is also evidenced by the popularity of conceptual maps, mind maps, and other similar methodologies to organize and summarize information. Our goal is to discover term relationships that can be used to construct conceptual maps or so called BisoNets.

The model we propose, tpf–idf–tpu, looks for pairs of terms that are associated in an individual document. It considers three aspects, two of which have been generalized from tf–idf to term pairs: term pair frequency (tpf; importance for the document), inverse document frequency (idf; uniqueness in the collection), and term pair uncorrelation (tpu; independence of the terms). The last component is needed to filter out statistically dependent pairs that are not likely to be considered novel or interesting by the user.

We present experimental results on two collections of documents: one extracted from Wikipedia, and one containing text mining articles with manually assigned term associations. The results indicate that the tpf–idf–tpu method can discover novel associations, that they are different from just taking pairs of tf–idf keywords, and that they match better the subjective associations of a reader.

1 Introduction

Documents are routinely characterized by their keywords, and keyword extraction is also a popular topic in text mining. Keywords certainly are useful, but they fail to describe relations between concepts in a document. In this chapter, we propose methods to mine characteristic term associations from unstructured documents in a given collection.

An example application is automatic generation of conceptual maps from news stories: such a map is a graph with terms or concepts as nodes and relations between them as edges. (Different flavors of such representations are known, e.g., as concept maps, mind maps, cognitive maps, and topic maps.) Conceptual maps are a well-known learning tool used to study and organize information, and one of our goals is to facilitate this process by automatic construction of rough conceptual maps.

M.R. Berthold (Ed.): Bisociative Knowledge Discovery, LNAI 7250, pp. 91–103, 2012.

In the context of creative information exploration and bisociative reasoning, such graphical representations are called BisoNets [1]. BisoNets can then be used to explore and discover novel information and unforseen connections between concepts.

As an example application, consider an online service that aggregates news stories from many sources and presents those to the user. Illustrating the novel association as a conceptual map together with suitable associations from the background knowledge provides a good overview of what is new in any particular story, and how it relates to existing information. As an example, consider the mining incident in 2010 in Chile, where 33 miners were trapped in a collapsed mine for more than two months before eventually being rescued via a newly drilled tunnel. In the first news stories, associations such as (*Chile, mine*), (*mine, collapse*) and (*miner, trapped*) were central. However, when more and more stories were written about the incident, these associations became part of the background. As the rescue operation advanced, new information became available about drilling and the tunnel, the rescue vessel to be used in it, the dates of the approaching final rescue operation, and eventually the success of the operation.

We are building such a prototype system, currently harvesting news from 7 online sources and with approximately 30 000 stories indexed so far. As an example, Figure 1 illustrates the essential associations, extracted with methods proposed in this chapter, from a news story published by The Washington Post[1] just before the lifting operation was to start. To highlight the news value of this story, the background associations relating the event to Chile, the capsule, etc. are not shown.

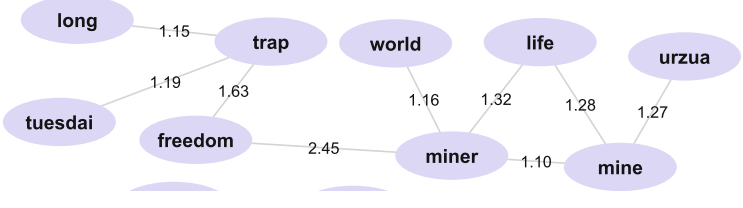


Fig. 1. Conceptual map of novel associations in a Washington Post news story "Chilean miners to begin emerging tonight" (Tue, Oct. 12, 2010). The miners had been trapped for over two months and were now about the be freed in an operation followed all around the world. Urzua is the name of the shift chief in the mine, a spokesman for the miners. Edge labels describe their importance.

Our goal is to extract interesting associations between terms in text document collections, to be presented, for instance, as simple conceptual maps or BisoNets. Roughly speaking, there are two different term association discovery tasks. The more standard one is discovering semantic similarities of terms, e.g.,

[1] `http://www.washingtonpost.com/wp-dyn/content/article/2010/10/12/ AR2010101203510.html?wprss=rss_world`

by their frequent co-occurrences. The other task, on which we focus in this chapter, is finding non-obvious, document-specific associations between terms. Note the strong contrast: in the latter task our aim is to discover novel associations between terms that are usually *not* related.

The remainder of the chapter is organized as follows: We will briefly review related work in Section 2. In Section 3 we propose a new method that finds exceptional relations in the sense that they are independent in the collection and specific to the document. Section 4 contains experimental results on two collections of documents: one extracted from Wikipedia, one containing text mining articles with manually assigned term associations. Section 5 contains concluding remarks and proposes further research on this topic.

2 Related Work

Conceptual maps, concept maps, mind maps, topic maps and many other similar formalisms exist for organizing and representing concepts and their relations as a graph. Many of them have been developed to be used as note taking and learning tools (see, e.g., [2]). Topic maps, on the other hand, are an ISO-standardized representation for interchange of knowledge. Unlike many of these techniques, we do not currently label edges by relation types. This could perhaps be done with information extraction methods (see below) after the associations have been discovered. We are not aware of methods for automatic, domain-independent construction of conceptual maps for documents in a given collection. We next review methods for finding various kinds of relations between terms or concepts.

There is abundant literature on finding statistical relations between terms. Most of the work is focused on discovering *semantically related terms*, such as *car* and *wheel*. Typically these techniques either use lexical databases and ontologies or measure co-occurrences of words, or combine these two. For instance, Hirst and St-Onge [3], as well as Patwardhan and Pedersen [4] measure semantic relatedness using *WordNet* as background knowledge. WordNet is a lexical database that consists of a thesaurus and several types relations between terms. WordNet-based similarity measures use path lengths between terms as the basis of relatedness. The Normalized Google Distance Measure (NGD) [5], in turn, uses Google search engine to measure the semantic relatedness of two terms. NGD has theoretical background in information theory, but in practice the idea is to compute the ratio of web pages where the terms occur independently to the pages where both of the terms occur. Latent Semantic Indexing (LSI) [6] goes beyond direct co-occurrence of terms, and uses singular value decomposition and reduction of matrix dimensions. Co-occurrence measures specifically aimed at *bisociation* are proposed by Segond and Borgelt [7]. They use keywords as the nodes of the BisoNet and focus on selecting appropriate edges between them. For the example application of producing conceptual maps, such semantic relations across documents are needed, and constitute an essential part of the background. The method proposed in this chapter addresses an opposite problem: find associations that are relatively specific to a document.

Our approach shares some mental similarity with *RaJoLink* [8] even though it works in a different setting. Given a collection of articles on some topic, RaJoLink starts by finding rare terms in it. The motivation is that these may be used to generate hypotheses about novel connections to other topics in further steps of the RaJoLink process. RaJoLink's emphasis is, however, on finding indirect relations of topics across documents, not on finding associations within documents.

The goal of *information extraction* is to extract certain structured information from textual documents (see, e.g., [9]). Information extraction methods are also routinely used to discover associations between terms. Examples include news story analysis (who did what, where and when) and automatic extraction of biomedical facts from scientific articles (which proteins interact, which gene contributes to which phenotype, etc.). While information extraction methods are tuned to look for specific types of facts (including relations), our goal is to be able to discover associations between arbitrary terms.

In *topic detection and tracking* the goal is to recognize events in news stories and to relate stories to each other [10]. In this task, information extraction is one of the key technologies. While we use news stories as an example application, our approach is largely complementary to topic detection and tracking: our emphasis is on relations between terms, both within stories (the novel associations looked for with methods introduced here) as well over several stories (semantic associtions in the background).

The technique we propose in this chapter is inspired by the well-known *tf–idf* (*term frequency–inverse document frequency*) keyword extraction method [11,12]. Term frequency $\mathrm{tf}(t, d)$ is the relative frequency of term t within a document d, and it measures how essential the term is for the document. The inverse document frequency $\mathrm{idf}(t)$ of term t measures, in turn, how specific the term is in the document collection. It is defined as the logarithm of the inverse of the relative number of documents that contain the term. Tf–idf for term t in document d is then the product $\text{tf-idf}(t, d) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t)$. Tf–idf and other methods to extract keywords (e.g., Keygraph [13]) have been highly successful in that task. However, they do not attempt to highlight associations between terms. Our aim is to discover association even if the individual terms are not important.

3 The tpf–idf–tpu Model of Important Term Pair Associations

We now propose and formalize a model for extracting important term associations from unstructured documents in a collection. The starting point is tf–idf [11,12], which we first generalize to pairs of terms. This generalization has, however, a serious shortcoming: term pair frequency and inverse document frequency do not sufficiently outrule possible correlation of the terms. We therefore add a third component, term pair uncorrelation.

We introduce two variants of the model that differ in the way the terms are paired in the documents. We use subscripts "sen" and "doc" to separate these

variants where necessary. The *sentence-level variant*, tpf–idf–tpu_{sen}, creates pairs from terms that occur in a same sentence. The *document-level variant*, tpf–idf–tpu_{doc}, pairs every term in the document with every other term in the document.

3.1 Term Pair Frequency (tpf) and Inverse Document Frequency (idf)

Term pair frequency $\text{tpf}_{\text{sen}}(\{t, u\}, d)$ is defined as the relative number of sentences s in document d that contain both terms t and u:

$$\text{tpf}_{\text{sen}}(\{t, u\}, d) = \frac{|\{s \in d \mid \{t, u\} \subset s\}|}{|\{s \in d\}|}. \tag{1}$$

The inverse document frequency $\text{idf}_{\text{sen}}(t, u)$ of term pair $\{t, u\}$ is the logarithm of the inverse of the relative number of documents in the given collection C that contain both terms in the same sentence:

$$\text{idf}_{\text{sen}}(t, u) = \log \frac{|C|}{|\{d \in C \mid \exists s \in d : \{t, u\} \subset s\}|}. \tag{2}$$

For the document-level variant, there are corresponding definitions of term pair frequency and inverse document frequency:

$$\text{tpf}_{\text{doc}}(\{t, u\}, d) = \min(\text{tf}(t, d), \text{tf}(u, d)), \tag{3}$$

where $\text{tf}(t, d)$ is the relative frequency of term t within a document d, and

$$\text{idf}_{\text{doc}}(t, u) = \log \frac{|C|}{|\{d \in C \mid \{t, u\} \subset d\}|}. \tag{4}$$

There is no natural direct measure of term pair frequency in a document. Following a common practice, we use the minimum of the frequencies of the two terms as the frequency of the pair.

3.2 Term Pair Uncorrelation (tpu)

Use of tpf–idf fails to recognize if there is a statistical (and possibly semantic) correlation between the terms. This is because tpf–idf only considers the joint occurrences of them, not if and how they occur without each other.

A pair that scores high on tpf–idf may be uninteresting for a number of reasons, but technically the reason usually is that the occurrence of one term (t) implies an occurrence of the other (u). Different instances of this problem include the following.

1. *Term t hardly ever occurs without term u.* For instance, articles that talk about "information retrieval" almost always mention "document", too.
2. *The two terms t and u occur roughly in the same set of documents.* For instance, "data mining" and "knowledge discovery" are roughly synonyms and obviously tend to occur in the same documents.

3. *Term u occurs in almost all documents.* For instance, "example" has a high document frequency. Paired with any less frequent term t, the tpf and especially idf scores can be high, but the association is trivial.
4. *Term t only occurs in few documents.* For instance, "tpf–idf–tpu" occurs so far only in this chapter. While associations with it are specific to this chapter, they are also trivial in a sense: any other term of this document would make a great pair with "tpf–idf–tpu", since the pair would trivially have an excellent idf score just because "tpf–idf–tpu" is so rare in a document collection.

In cases 1 and 2, the association between t and u is real but not document-specific, and therefore it should be part of the background. Cases 3 and 4 are trivial and therefore not interesting.

To rule all the above-mentioned cases out, we add a third component to the model: term pair uncorrelation, or tpu. We define tpu in terms of the relative amounts $\mathrm{r}(v)$ (where $v = t$ or u) of co-occurrences in the document collection:

$$\mathrm{r}_{\mathrm{sen}}(v) = \frac{|\{d \in C \mid \exists s \in d \text{ s.t. } \{t, u\} \subset s\}|}{|\{d \in C \mid v \in d\}|}. \tag{5}$$

The value of $\mathrm{r}(t)$ is 1 if t and u always co-occur, 0 if they never co-occur, and 0.5 if u co-occurs in half of the documents in which t occurs.

We prefer that both terms occur often independently, i.e., that both $\mathrm{r}(t)$ and $\mathrm{r}(u)$ are small. To measure this, we simply take their maximum. (Alternative measures for tpu include Jaccard index and Tanimoto coefficient. We prefer the measure based on $\max(\mathrm{r}(t), \mathrm{r}(u))$, however, since it more strongly requires that both terms also occur independently.)

In order to have a tpu measure that has larger values for the preferred situations, we define tpu as

$$\mathrm{tpu}(\{t, u\}) = \gamma - \max(\mathrm{r}(t), \mathrm{r}(t)), \tag{6}$$

where γ tunes the relative importance of the tpu component, $\gamma \geq 1$. Smaller values of γ give tpu more weight. An analysis of the effects of γ is outside the scope of this chapter. In our experiments we use $\gamma = 2$ based on some preliminary experiments.

For document-level analysis, we define $\mathrm{r}_{\mathrm{doc}}(v)$ as

$$\mathrm{r}_{\mathrm{doc}}(v) = \frac{|\{d \in C \mid \{t, u\} \subset d\}|}{|\{d \in C \mid v \in d\}|}. \tag{7}$$

Finally, $\text{tpf–idf–tpu}(\{t, u\}, d)$ of term pair $\{t, u\}$ in document d is defined as the product of the three components defined above:

$$\text{tpf–idf–tpu}(\{t, u\}, d) = \mathrm{tpf}(\{t, u\}, d) \cdot \mathrm{idf}(\{t, u\}) \cdot \mathrm{tpu}(\{t, u\}).$$

4 Experiments

In the following subsections we experimentally evaluate the performance of the tpf–idf–tpu model. We contrast the discovered term pairs to keywords obtained

using tf–idf, and we also compare the sentence and document-based variants to each other.

Unfortunately, we are not aware of existing data sets with documents and corresponding conceptual maps, so we have to resort to other test methods. We use two different test settings.

In the first setting (used in Sections 4.1 and 4.2) the document collection consists of 425 articles on everyday life (and its subtopics), obtained from the Wikipedia Selection for Schools[2]. We use this document collection to compare the sets of term pairs produced by the different variants.

In the second test setting (Section 4.3), we created a collection of annotated text mining documents. One of the authors of this chapter manually annotated 23 documents with term associations that he considered most descriptive for the topic of each document. The document collection additionally contains another 15 text mining articles, so the total size of the collection is 38 documents. The manually assigned 229 term pairs were considered equally important and thus not ordered nor weighted in any way. Subjective annotation of key terms (or term pairs, in our case) is criticized in the literature, as the background, interests and viewpoint of the annotator affect what he or she considers to be relevant [14]. With this precaution in mind, we believe that such an evaluation can give indications of the performance of the method.

In both settings, the documents were preprocessed by removing stopwords and by stemming with Porter stemmer [15]. In addition, automatic multiword unit extraction was performed with Text-NSP program [16] using log-likelihood measure. Consecutive sequences of two terms, or *bigrams*, that got log-likelihood score of 70 or higher were treated as one term.

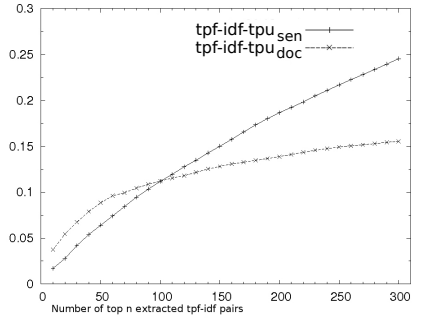
The goal of these tests is to give a first evaluation and illustration of the potential of the method. More systematic experiments on different data sets are left for future work.

4.1 Tpf–idf–tpu vs. tf–idf

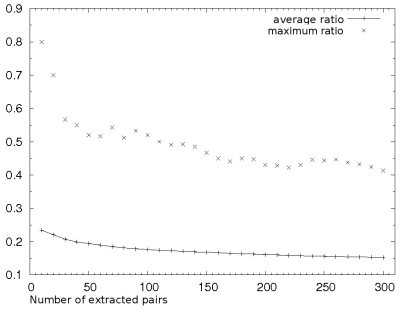
Let us first address the question if and how different the results of term pair extraction are from single keyword extraction. To study this, we performed the following experiment with the everyday document collection.

First, n best tpf–idf–tpu term pairs were extracted from each document. Then the pair structure was ignored and we simply considered the set of terms in these top pairs. Then, an equal number of top tf–idf terms were extracted from each document. As an evaluation measure, we used the ratio of the number of terms produced by both methods divided by the total number of terms produced by the methods. The ratios were computed for a wide range of values of n, the number of top pairs to be picked in the first phase. For each n the average of the ratios from all documents was computed. The results are shown in Figure 2(a) as a function of n.

[2] `http://schools-wikipedia.org/`, downloaded in 2010.



(a) Average ratio of identical terms to all terms in the top n results of tf–idf and the tpf–idf–tpu variants.



(b) The ratio of identical pairs to all extracted pairs in the top n pairs extracted by the two tpf–idf–tpu variants.

Fig. 2. Overlap of results from tpf–idf–tpu variants and tf–idf

The results of this experiment clearly show that the terms extracted by the tpf–idf–tpu and tf–idf methods differ considerably, even with large numbers of extracted pairs. The tpf–idf–tpu method does not just create pairs of top ranking tf–idf terms, but actually does extract other relations. At ten top pairs, the ratio of identical tpf–idf–tpu_{sen} and tf–idf terms is only about 2% on average and rises to approximately 25% at 300 pairs. The ratio of identical tpf–idf–tpu_{doc} and tf–idf terms in top ten pairs is about 2%, and rises to about 15% at 300 pairs.

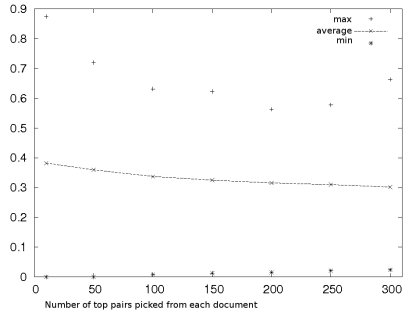
4.2 Sentence vs. Document-Level tpf–idf–tpu Methods

We next compare the sentence and document-level tpf–idf–tpu directly to each other. We will consider three related but different aspects: (1) how similar are the term pairs chosen by the methods, (2) how similar are the terms in the pairs chosen by the methods, and (3) are the pairs dominated by a small number of terms.

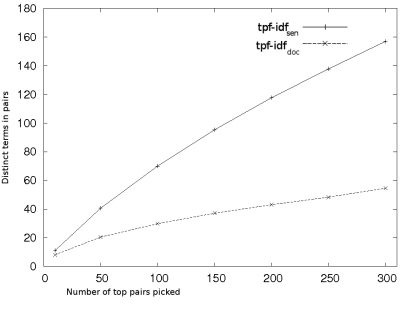
First, the similarity of tpf–idf–tpu_{sen} and tpf–idf–tpu_{doc} is examined by comparing their top scoring pairs. This is done by extracting top n pairs with each method, and computing the ratio of identical pairs in the top n pairs to the total number of pairs, that is, to $2 \cdot n$. To combine the ratios yielding from different documents, the average, minimum and maximum of the ratios were taken. The results are shown in Figure 2(b) as a function of n, the number of extracted top pairs. The minimum ratio was zero for all n.

The experiment indicates that the top pairs produced by tpf–idf–tpu_{sen} and tpf–idf–tpu_{doc} differ considerably. The average ratio is slightly higher for small numbers of extracted pairs. This indicates that the highest ranking pairs tend to be slightly more similar. At top ten term pairs extracted by tpf–idf–tpu_{sen} and tpf–idf–tpu_{doc}, the average ratio is about 25% and maximum ratio is about 80%. At 300 top pairs the ratio of identical pairs lowers to about 15% and the maximum ratio to about 40%.

Next, the ratio of identical terms in the top pairs produced by tpf–idf–tpu_{sen} and tpf–idf–tpu_{doc} was studied. The motivation for this experiment was to see if the methods generate the pairs from a similar set of terms but pair them in different ways. The experiment was performed by selecting top n pairs for a document by both tpf–idf–tpu_{sen} and tpf–idf–tpu_{doc} methods. Then we again computed the ratio of the number of identical terms in the top n pairs divided by the total number of distinct terms in the pairs. Like in the previous experiment, the average of these ratios from different documents was taken. In addition to the average ratio, the minimum and maximum ratios are considered (Figure 3(a)).



(a) The ratio of identical terms to all terms in the top n pairs extracted by the two tpf–idf–tpu variants.



(b) The number of distinct terms in top n pairs extracted either with tpf–idf–tpu_{sen} or with tpf–idf–tpu_{doc}.

Fig. 3. Overlap of results from tpf–idf–tpu variants and tf–idf, and internal variability in tpf–idf–tpu results

The ratio of identical terms in the pairs is about 40 percent on average and almost 90 percent at maximum when comparing top ten pairs. The ratios of identical terms in Figure 3(a) are clearly higher than the ratios of identical pairs in Figure 2(b), although on average the ratio is not very large.

Next we consider the number of distinct terms in the pairs produced by tpf–idf–tpu. The goal is to see if the top pairs are dominated by a small set of distinct terms. For this test, the top n pairs were picked from each document and the average number of distinct terms was computed over the documents (Figure 3(b)).

The number of distinct terms is relatively low for both of the methods. Especially pairs produced by tpf–idf–tpu_{doc} are dominated by a small set of terms. For top ten pairs the number of distinct terms is about ten on average for both tpf–idf–tpu_{sen} and tpf–idf–tpu_{doc}. At 300 top pairs the number of distinct term rises to about 160 for tpf–idf–tpu_{sen} and to about 60 for tpf–idf–tpu_{doc}. In comparison, 25 terms is the minimum number of terms to produce 300 pairs; in tpf–idf–tpu_{doc} there are about 60 terms on average that occur in the 300 top pairs.

It is not clear from these results if a smaller or larger number of distinct terms should lead to a better result. It is possible that the smaller term set used by

tpf–idf–tpu_{doc} contains less noise than the larger set extracted by tpf–idf–tpu_{sen}. On the other hand, it could also miss relevant terms and term pairs.

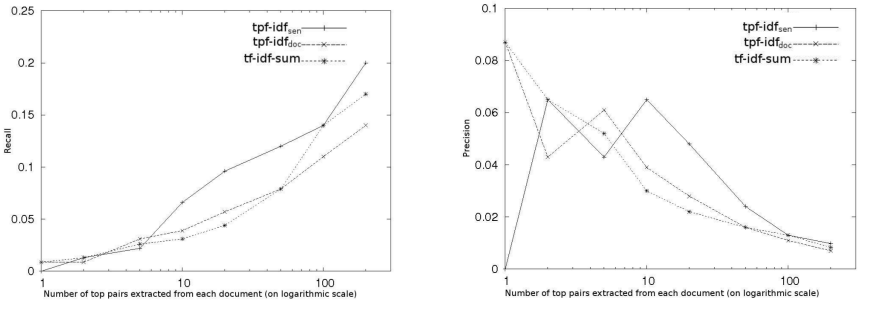
4.3 Comparison of tpf–idf–tpu and tf–idf Using Annotated Test Set

We now move to experimental tests with the other document collection, text mining articles, and compare the results of the methods against pairs annotated by hand. As a simple baseline method, we used tf–idf to rank pairs of terms by simply taking the sum of the terms' individual tf–idf scores.

For each method, precision and recall were computed at several points in range of $n = 1$ to 300 top pairs per document. Precision is the ratio of extracted annotated pairs to n, the total number of pairs chosen, where "annotated" means that the pair was among ones manually assigned to the document. Recall is the ratio of extracted annotated pairs to all annotated pairs. In an optimal situation both precision and recall would be high for the extracted top pairs, meaning that in the top pairs there would be no non-key pairs and no key pairs would be missing either.

There were 229 annotated pairs in total. From those, 66 pairs were out of reach for the tpf–idf–tpu_{sen} method since the terms never co-occurred in the same sentence. Because of this, extraction of all possible pairs only yields recall of 0.71 for tpf–idf–tpu_{sen}. On the other hand, the number of term pairs per document varied from 3 561 to 55 552 for tpf–idf–tpu_{sen} and from 118 341 to 3 386 503 for tpf–idf–tpu_{doc} and tf–idf–sum.

The results for recall and precision (Figure 4) indicate the following. First, due to the small number of documents, the results for $n = 1$ to 5 are very noisy, and it is difficult to observe systematic differences between any of the three methods. Then, however, for $n = 10$ to 100 extracted pairs, the sentence-based method consistently outperforms the other two, in terms of both precision and recall. The document-based method has a slight systematic edge over the tf–idf-baseline in the mid-range. For $n \geq 100$, the tf–idf-baseline in turn outperforms the document-based method.



(a) Recall. (b) Precision.

Fig. 4. Recall and precision at different numbers of extracted pairs

The recall and precision values may seem low. Notice, first, that the setup of this experiment differs from the usual precision and recall experiments in document retrieval. In this experiment only the annotated associations are classified relevant; all the other pairs are implicitly classified as irrelevant even though they are not inspected in any way for relevance or novelty. It is thus possible that there are pairs that could be considered relevant for the document even though they were not selected as key pairs in the manual annotation. Second, consider the extreme challenge in the task: on average, 10 pairs were manually extracted from each document, whereas the number of different pairs per document ranges approximately from 3 500 to 3 400 000, depending on the method. In other words, the fraction of manually tagged pairs ranges from 0,0000003 to 0,003. Compared to this scale, the numbers are high.

According to the results, we believe that tpf–idf–tpu_{sen} has great potential to discover important associations between terms. The document-based variant performs less consistently. Since the two variants find largely different pairs, it will be an interesting topic for future research to try to combine their best properties.

5 Conclusion

We have proposed to discover novel associations of terms in unstructured documents, and to use these to summarize the key concepts and relationships of the documents. A term pair has a novel association in a document if the pair is frequent in it (tpf), specific to it (idf), and uncorrelated in the document collection (tpu). The proposed method, tpf–idf–tpu, is a generalization of tf–idf to pairs of terms, with the tpu component added to avoid statistically related pairs of terms.

We proposed two variants of tpf–idf–tpu: the document-level version checks if the terms co-occur within a document, while the sentence-level variant only considers the terms to co-occur if they are in the same sentence. For comparison, we also implemented a simple tf–idf-based method that outputs pairs of keywords.

We experimentally observed that tpf–idf–tpu produces pairs (and terms) significantly different from tf–idf. The sentence and document-based variants also produced results quite different from each other. In a recall/precision analysis with a smaller, manually annotated set of documents, the tpf–idf–tpu_{sen} variant based on sentence-level pairing of terms performed clearly better than the other methods when 10-100 term pairs were extracted per document. For smaller numbers of extracted associations, the results are noisy and inconclusive. Systematic experiments on different data sets are a topic for future work.

We are currently building an experimental online news summary system to try out how an incremental version of tpf–idf–tpu manages to identify and summarize the novelties in news stories and to visualize them as simple conceptual graphs. For this task, semantic associations should also be extracted and visualized as background knowledge.

We plan to apply graph mining and bisociation methods on the conceptual graphs, e.g., to discover more distant relationships between concepts. For such use, it could be useful to keep the tpu score separate from the tpf–idf scores, and allow the graph mining algorithms to consider the strength of the link ($\text{tpf} \cdot \text{idf}$) and its unobviousness (tpu) separately.

Acknowledgment. This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland (Grant 118653) and by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898.

References

1. Kötter, T., Berthold, M.R.: From Information Networks to Bisociative Information Networks. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 33–50. Springer, Heidelberg (2012)
2. Novak, J.D., Cañas, A.J.: The theory underlying concept maps and how to construct them. Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition (2008)
3. Hirst, G., St-Onge, D.: Lexical chains as representations of context for the detection and correction of malapropisms. In: Fellbaum, C. (ed.) WordNet: An Electronic Lexical Database, pp. 305–332. MIT Press (1998)
4. Patwardhan, S., Pedersen, T.: Using WordNet based context vectors to estimate the semantic relatedness of concepts. In: EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together, pp. 1–8 (April 2006)
5. Cilibrasi, R.L., Vitányi, P.M.: The Google similarity distance. IEEE Transactions on Knowledge and Data Engineering 19(3), 370–383 (2007)
6. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)
7. Segond, M., Borgelt, C.: Selecting the Links in BisoNets Generated from Document Collections. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS, vol. 7250, pp. 56–67. Springer, Heidelberg (2011)
8. Petrič, I., Urbančič, T., Cestnik, B., Macedoni-Lukšič, M.: Literature mining method RaJoLink for uncovering relations between biomedical concepts. Journal of Biomedical Informatics 42(2), 219–227 (2009)
9. Cowie, J.R., Lehnert, W.G.: Information extraction. Communications of ACM, 80–91 (1996)
10. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 194–218 (1998)
11. Spärck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28, 11–21 (1972)

12. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM 18(11), 613–620 (1975)
13. Ohsawa, Y., Benson, N.E., Yachida, M.: Keygraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In: ADL 1998: Proceedings of the Advances in Digital Libraries Conference, vol. 12. IEEE Computer Society, Washington, DC (1998)
14. Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of Research Development 2(2), 159–165 (1958)
15. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
16. Banerjee, S., Pedersen, T.: The design, implementation and use of the ngram statistics package. In: Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, pp. 370–381 (2003)