Designing topic shifts with graphs

Martín I. Rezk and Laura Alonso i Alemany

FaMAF, Universidad Nacional de Córdoba
Córdoba, Argentina,
rm1@hal.famaf.unc.edu.ar,alemany@famaf.unc.edu.ar

Abstract. We present CHESHIRE, a recommendation system to help in building reading lists for topic shifts. Given a document collection, a starting topic and a target topic (expressed by keywords), CHESHIRE recommends the sequence of documents that bridges the gap between input and target topics with the smallest difference in content between each document. To do that, the document collection is represented as a graph, where documents are nodes related by weighted edges. Edges are created whenever a set of words is shared by two documents. In this paper we present experiments with different methods for choosing words that create edges, and the weights to be assigned to each edge. Results are evaluated by comparison with a dummy baseline and with a manually created gold standard.

1 Introduction and Motivation

"Would you tell me, please, which way I ought to go from here?"
"That depends a good deal on where you want to get to," said the Cat.
"I don't much care where –" said Alice.
"Then it doesn't matter which way you go," said the Cat.
"– so long as I get somewhere," Alice added as an explanation.
"Oh, you're sure to do that," said the Cat, "if you only walk long enough."

From Lewis Carroll's Alice in Wonderland

From this side of the mirror, people may have different needs than Alice's. The work presented here aims to address one of these needs, namely, to reduce uncertainty at the starting point of a shift in your research topic.

Given the enormous amount of information (let's call it documents) available today, starting a new topic is appalling. Given the evident lack of talking cats in research environments, we found an alternative to Lewis Carroll's way.

The usual way to advance in bibliographical research is by asking one's advisor. The effectiveness of this method lies in the fact that advisors know:

1. students' background,
2. targeted knowledge, and
3. what papers bridge the gap between the two previous items.

In this paper we present CHESHIRE, a system that automatically suggests a sequence of papers to be read in order to get from a given background to a targeted knowledge. Both background and target are manually provided to the system by way of keywords.

Our starting hypothesis is that documents in a given collection may share part of their content but not necessarily all of it. So, it is possible to establish a sequence of documents so that the content in a document is not entirely unseen in preceding documents but also not entirely covered by them. This configuration provides a smooth shift from one topic to another, which is precisely what the system presented here intends.

In order to find the optimal sequence of documents, the structure of a document collection is represented as a graph, where nodes are documents and edges are relevant words linking those documents. This representation lies in the assumption that the content of documents can be represented by relevant words. It is known that word forms are highly ambiguous and thus provide an error-prone representation of content, but this noisy representation can be counterbalanced by exploiting the redundancy of natural language, as in most IR applications.

The rest of the paper is organized as follows. In the next section we provide the basic concepts on which the rest of the paper is based. Section 3 describes the architecture of the system presented here, and in Section 5 we analyze the results of the system, and obtain some interesting insights about the properties of document collections that will be used in future developments of the system. We finish with some conclusions and future work.

2 Background

2.1 Similarity between documents

When we face the problem of finding a sequence of documents joining two topics, the first decision to be taken is the criterion to decide when two documents are semantically related.

In the first place, we take the intuitive notion of similarity given by Lin (1998):

- The similarity between A and B is related to their commonality. The more commonality they share, the more similar they are.
- The similarity between A and B is related to the differences between them. The more differences they have, the less similar they are.

Many methods have been proposed to calculate the similarity between two objects. Following Lin, a general formula to calculate similarity would be:

$$I(common(A, B))$$

where $common(A, B)$ is a proposition that states the commonalities between A and B; and $I(s)$ is the amount of information contained in a proposition s.

In order to establish what A and B have in common, we exploit *relevant words* in the text. By relevant words we mean the subset of words that best represent the document's semantics. But how can they be found?

The simplest approach is to take the more frequent words as relevants. Various refinements to this basic method can be applied, like removing stopwords, calculating weighted forms of frequency, like $tf * idf$ (Salton and McGill 1983), etc. Words have also been considered relevant if they occur in prominent locations, like title, abstract, keyword section, etc.

But frequency-based characterization does not provide a representation of relevance accurate enough. Indeed, important words may be less frequent than others.

That's why, and considering that natural language can be seen as a small world network (Newman 2000), we applied the concept of Capital and Benefit proposed by Licamele et al. (2005). The basic concept is very simple: given a set of words that we positively know are important in a document, we assume that words that occur very frequently with them will also be important. The more important words a given word occurs with, the more important the word itself will be. For our purposes, important words will be those that best represent the semantic content of the document.

Summarizing:

W	a set of words $W = w1, ..., wn$
D	a set of documents $D = d1, ..., dm$
$C(w_i, w_j)$	gives the probability of co-occurrence of w_i with w_j (in the original paper, "*is friends with*")
$Imp(d_k)$	is the set of words important to document d_k (in the original paper, "*organizers*")
$I(C(w_i, w_j))$	is true if $C(w_i, w_j)$ is above a certain threshold

Definition 1. Social Capital. *The social capital of a word w_i in a document d_k is the number of important words with whom the word co-occurs:*

$$SC(w_i, d_k) = \sum_{w_j \in Imp(d_k)} I(C(w_i, w_j))$$

Definition 2. Social Capital Ratio. *The capital ratio of a word w_i in a document d_k is the proportion of important words with whom w_i co-occurs*

$$SCR(w_i, d_k) = \frac{\sum_{w_j \in Imp(d_k)} C(w_i, w_j)}{|Imp(d_k)|}$$

2.2 Document Space as a Graph

Once we have settled the criterion to decide when two documents are semantically related, the relations between document will be represented as a graph.

A graph representation has many advantages. It is more understandable, it can be represented visually, and, in addition, the sides can be weighed indicating the degree of the relation between documents. Besides, we can exploit properties

of the kind of graphs that emerge in document collections, the so-called small world graphs (Newman 2000), like the existence of hubs. For example, given a corpus containing disjoint topics, the $tf * idf$ can be more accurate if it is not calculated in the entire corpus, but rather within each connected component of the graph separately (or more specifically in SWN, in hubs). This is so because hubs can be seen like clusters with a thematic in common.

In order to reflect that the similarity between documents is not symmetrical, directed graphs are used.

But the most useful property of a graph representation for our purposes is that it helps to find the shortest and ligthest path from one vertex to another. One of the most popular algorithms to find the best path in a graph is Dijkstra's (Dijkstra 1959). Dijkstra's algorithm is based in the foundation of the optimality principle: if the shortest path between vertices u and v passes through the vertex w, then the part of the way that goes from w to v must be the shortest path between all the ways that go from w to v.

3 Architecture

The input to the system is:

- a document collection,
- a set of keywords constituting the starting topic (reader's background), and
- a set of keywords constituting the target topic.

3.1 Preprocessing

Documents in the collection are enriched by a parsing module, which identifies:

- Word **lemmas** in the document, with their **frequency** of occurrence. Lemmatization is by FreeLing (Atserias et al. 2006), stopwords are removed.
- Relevant **layout components**: title, abstract, keywords, conclusions, references.

3.2 Weighting

Then, a weighting module assigns each word a score according to their relevance to characterize the content of the document. We have experimented with different modes for weighting words:

Frequency the score of a word is directly proportional to its probability of occurrence in the document.

tf*idf the score of a word is directly proportional to its $tf * idf$.

Relevant Words words are assigned a higher score if they occur in relevant layout components.

Capital and Benefit given a set of relevant words (determined by any other method), the score of a word (or *Capital Ratio*) is directly proportional to its probability of co-occurrence with relevant words.

3.3 Creating the Graph

Each document is characterized by its k highest scoring words. Then, an inverse dictionary is created, where each characterizing word is related to the set of documents that it characterizes, together with its score. Building a graph from this inverse dictionary is trivial. Nodes are documents, and edges between each pair of nodes are created by those words that characterize both documents. Edges are decorated with the cost of the transition between documents:

$$Cost_{d,d'} = 1000 - \sum_{cw_{d,d'}} score(cw)$$

where

d, d'	are the pair of documents to be related
$cw_{d,d'}$	is a word characterizing both documents d and d'
$score(cw)$	is the score assigned to the word

Thus, the more characterizing words documents share, the lower the cost of the transition between them. In addition, the cost of the transition is also lowered by words with high score. This is a direct consequence of the hypothesis that the more characterizing words shared by a pair of documents, the more content they share, and so it is easier to understand one having read the other.

4 Experiments

For the experiments reported here, the system has worked with the following parameters:

- characterizing words: k was settled to 10, so for each document, we selected the 10 words with highest score by any of the weighting methods described above.
- strong edges: only links between documents consisting of more than one word are considered.
- $tf * idf$: the inverse document frequency is calculated in the whole document collection, because it belongs to a homogeneous topic. In case different topics are contained within the collection, the inverse document frequency can be calculated independently within topics.

We obtained the following runs:

Frequency Baseline the 10 words with highest probability of occurrence in the document are considered characterizing.

Frequency + Layout the contribution of words in relevant layout sections (title, keywords, abstract, references and conclusions) is quantified as $score(cw) = P(w) * 0.1$, whereas the contribution of words not occurring in relevant layout sections is quantified as $score(cw) = P(w) * 0.001$.

tf∗idf the 10 words with highest $tf * idf$ are considered characterizing.

Capital and Benefit words in the title are considered relevant, and the 10 words with highest probability of co-occurrence with them are considered relevant. Probability of co-occurrence with a relevant word (capital ratio) is computed as the probability of occurring in a sentence where a relevant word occurs. Note that, computing capital this way, the words in the title themselves have a very high score.

Capital + tf∗idf the score of words is calculated as $score(cw) = tf * idf_{cw} * (CapitalRatio_{cw} + 1)$.

4.1 Corpus and Gold Standard

We have created a corpus of 31 documents consisting of research papers in the Computational Linguistics domain, with a total of 81,287 words, ranging from documents of almost 10,000 words to a document of 416 (median is 4000 words). The total number of lemmas in the corpus was 18,156, of which 64% occur only in one document, 14% in two documents, 6% in three documents and 16% occur in four or more documents.

The documents in the collection have been selected as configuring three disjoint paths of the kind given by the system, described in Figure 1. These paths are taken as the *gold standard* to evaluate the performance of the system.

path 1
input named entity recognition
output boosting
5 documents, 4 steps (1 doc → 2 docs → 1 doc → 1 doc)
path 2
input lexical ontologies, EuroWordNet, SUMO
output reasoning for question answering
9 documents, 4 steps (2 docs → 2 docs → 3 docs → 2 docs)
path 3
input automatic text summarization
output discourse relations
4 documents, 3 steps (1 doc → 2 docs → 1 doc)

Fig. 1. Paths of topic shift defined manually in the corpus, taken as the *gold standard* for evaluation of CHESHIRE.

Additionally, 10 unrelated documents of the same topic (Computational Linguistics) were included in the corpus, in order to make the collection more realistic.

Documents were transformed from pdf to plain text by unix utility *pdf2txt*. Errors due to faulty conversion have not been quantified but they do not seem to have much impact in the representation of documents in terms of keywords.

Lemmatization is the second source of noise for the preprocessing of the corpus. The main errors in lemmatization affect multiword expressions, which are not recognized as such.

5 Analysis of Results

We provide two different analysis of results: first, we study differences in characterizing words in each of the approaches. Then, we describe how these approaches perform as compared with the gold standard, in terms of accuracy in path retrieval. In general, the system has 100% precision in recovering paths, but recall is low (around 50%).

In this analysis, the baseline is provided by the system run where characterizing words are the 10 most frequent words in each document. The gold standard has been created manually as described in the previous section.

5.1 Characterizing Words

In Table 1 we can see the 10 words most frequently selected as characterizing for documents in each run. The first column represents the most frequent words in the corpus, having removed stopwords. As can be expected, the 10 words most frequently chosen as characterizing by the baseline overlap highly with the most frequent words in the corpus (60%). What is not so expectable is that the $tf*idf$ method selects almost the same set of words.

Taking into consideration layout (approaches *frequency+layout, Capital* allows different words to be chosen (*semantic, language*). The combination that has the largest proportion of words not within the 10 most frequent in the corpus is *Capital+tf*idf*, with only 30% of words within the 10 most frequent in the corpus.

In Table 2 we can see that there is a very high correspondence between words most frequently chosen as characterizing and words most frequently occurring in the edges of the graph.

Finally, in Table 3 we can see the words most frequently occurring in those edges of the graph that are selected to connect the document retrieved as the best representative of the input topic and the document retrieved as the best representative of the target topic. No words are given in the frequency baseline because it failed to build a path between the two documents. It can be seen that the *Capital* method, and the method *Capital + tf*idf* to a lesser extent, are the ones with more words in the edges. This indicates that the capital method is describing documents more exhaustively than frequency-based methods.

5.2 Accuracy in path retrieval

The results of document retrieval do not differ significantly across the different runs of the system. In all the cases, all documents retrieved correspond to a document in the gold standard path, thus we have 100% precision for all runs.

	baseline	freq. + layout	$tf * idf$	Capital	Capital + $tf * idf$
word	system	text	system	system	word
text	word	word	text	text	text
set	text	system	word	word	**answer**
system	model	model	model	model	**wordnet**
model	one	**language**	one	**language**	**semantic**
one	**wordnet**	**wordnet**	set	one	model
example	set	set	**relation**	set	**name**
result	**relation**	**semantic**	**name**	**semantic**	**generation**
feature	**question**	**question**	**question**	**build**	**question**
data	**name**	**name**	**answer**	**name**	**discourse**

Table 1. Words most frequently selected as characterizing (the first column represents the most frequent words in the corpus, having removed stopwords).

	baseline	freq. + layout	$tf * idf$	Capital	Capital + $tf * idf$
word	system	system	system	system	word
text	word	set	text	text	text
set	model	word	word	word	**semantic**
system	one	**algorithm**	one	model	model
model	**relation**	**language**	model	**language**	**answer**
one	**question**	**relation**	set	one	**name**
example	feature	data	**relation**	set	**generation**
result	**answer**	sense	example	**semantic**	**question**
feature	**wordnet**	**answer**	**language**	**build**	**wordnet**
data	set	**question**	**name**	example	**sense**

Table 2. Words most frequently occurring in edges.

Considering the ambiguity in natural language and the fact that all documents in the collection share a high number of words, precision of 100% is a very good result.

However, steps in the gold standard path are often skipped by the system, which results in an average recall about 50% (50% for path 1 and 2, 66% for path 3). Note that this is the shortest possible path between a document representing the input topic and a document representing the target topic, instead of the smoothest. This is a clear bias of Dijkstra's algorithm, which must be taken into account for future versions of the system.

It must be noted, however, that the frequency baseline fails to build a path from the document retrieved as the most representative of the input topic to the document retrieved as the target topic. Moreover, capital methods characterize documents much more exhaustively, which is reflected in the fact that in some cases they have produced sequences of documents of length 3 instead of 2. Thus, it seems clear that these methods would benefit dramatically from a graph traversal algorithm that priorizes smoothness to shortness.

	baseline	freq. + layout	$tf * idf$	Capital	Capital + $tf * idf$
word		1 term	2 *text*	3 *text*	1 web
text		1 structure	1 web	2 work	1 vote
set		1 structural	1 tree	2 wordnet	1 tree
system		1 sentence	1 textual	2 build	1 textual
model		1 relation	1 *system*	1 tree	1 *text*
one		1 query	1 structure	1 structure	1 semantic
example		1 *one*	1 semantic	1 *set*	1 rhetorical
result		1 method	1 search	1 rhetorical	1 ontology
feature		1 lexical	1 rhetorical	1 relation	1 name
data		1 document	1 provide	1 question	1 *model*
		1 discourse	1 ontology	1 proper	1 extraction
		1 clause	1 literature	1 ontology	1 entity
		1 boosting	1 exist	1 noun	1 discourse
		1 answer	1 discourse	1 method	1 classifier
				1 literature	1 boosting
				1 language	1 adaboost
				1 information	
				1 exist	
				1 discourse	
				1 database	
				1 answer	

Table 3. Words occurring (and number of occurrences) in edges in the path connecting the document retrieved as the best representative of the input topic and the document retrieved as the best representative of the target topic.

6 Conclusions and Future Work

We have presented CHESHIRE, a system that, given a document collection, a starting topic and a target topic (expressed by keywords), recommends the best sequence of documents that bridges the gap between input and target topics.

The document collection is represented as a graph, where documents are related by characterizing words. Different methods for finding characterizing words have been evaluated, and we have found that a method based on small world networks performs best. However, differences in characterization of the documents are not reflected in the final results of the system, probably because of the size of the evaluation corpus and the bias of the path traversal algorithm towards shorter paths. Thus, future work includes increasing the amount of evaluation material, and also comparing different graph traversal algorithms.

A further development of the work presented here is to use a multigraph instead of a simple graph, so that each pair of documents can be linked by more than one edge, each edge corresponding to a single word.

Another improvement of the system will be using a better lemmatization tool, that is able to identify multiword expressions that are crucial to characterize the content of technical documents.

References

[Atserias, Casas, Comelles, González, Padró, and Padró 2006] Atserias, J., B. Casas, E. Comelles, M. González, L. Padró, and M. Padró (2006). Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06)*.

[Dijkstra 1959] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 269–271.

[Licamele, Bilgic, Getoor, and Roussopoulos 2005] Licamele, L., M. Bilgic, L. Getoor, and N. Roussopoulos (2005). Capital and benefit in social networks. In *Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD-2005)*.

[Lin 1998] Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of ICML-98*.

[Newman 2000] Newman, M. (2000). Models of the small world. *Journal of Statistical Physics 101*, 819.

[Salton and McGill 1983] Salton, G. and M. J. McGill (1983). *Introduction to modern information retrieval*. McGraw-Hill.