# DISCO: A System Leveraging Semantic Search in Document Review

**Ngoc Phuoc An Vo, Fabien Guillot, Caroline Privault**
Xerox Research Centre Europe
Meylan, France
{an.vo, fabien.guillot, caroline.privault}@xrce.xerox.com

## Abstract

This paper presents Disco, a prototype for supporting knowledge workers in exploring, reviewing and sorting collections of textual data. The goal is to facilitate, accelerate and improve the discovery of information. To this end, it combines Semantic Relatedness techniques with a review workflow developed in a tangible environment. Disco uses a semantic model that is leveraged on-line in the course of search sessions, and accessed through natural hand-gesture, in a simple and intuitive way.

## 1 Introduction

Complex information seeking tasks frequently involve exploratory search activities. Although they can be characterized in many ways (see (Wildemuth and Freund, 2012), or (Marchionini, 2006) where they are grouped into "Learn" and "Investigate" activities), the next aspects are frequently used to describe exploratory scenarios: a) task description is ill-defined: it is broad or under-specified, or on the contrary multi-faceted; b) task is dynamic: relevance, object, targets evolve over search time; c) Information need is ill-defined: initial searchers' knowledge may be insufficient or inadequate and will also evolve over time. A remarkable consequence is that searchers have more latitude in directing their search. They can follow mixed strategies of searching, where they alternate between exploration phases and lookup/iterative phases. In the latter, items are systematically searched or reviewed (e.g. by attribute or simple keyword), whereas during exploration the search is expanded to new data, sources or domain of information, or to the development of new search criteria or strategies.

The development of search tools and interfaces to support exploratory search activities present a range of design challenges (e.g visualization, interaction, or relevance feedback). Recently search interfaces have been designed on multi-touch devices (smart phones, tablets (Klouche et al., 2015) or large surfaces). However, user studies reveal the need for search systems to increase the level of explorative search versus iterative search; otherwise, users tend to actually engage in exploring and learning from the data set, but in a rather limited way despite the availability of advanced UI layout and features. Search tools should then specifically encourage users to engage into exploratory phases, and facilitate the switch between lookup and exploratory phases.

Disco is a prototype developed to support knowledge workers in exploring, reviewing and sorting collections of textual data. We present how semantic relatedness is leveraged to sustain explorative search and increase information discovery. Our system is targeted at every user including non-technical users and is not domain-specific.

## 2 System Description

### 2.1 Disco Functionalities and Tangible User Interface

Disco combines a tangible user interface (TUI) with machine learning and advanced search capabilities (Privault et al., 2010; Xerox, 2010). At session startup, the user loads a collection of documents that is displayed on a touchscreen in a "Wall view": each document is represented by a tile on the wall. The user can explore the data set by using unsupervised text clustering, ML text categorization, automatic term

*Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 64–68, Osaka, Japan, December 11-17 2016.

extraction and keyword-based filtering. When the user locates a subset of documents that seem worth further reviewing, the subset can be sent to a dedicated area called the "Document Dispenser" (DD) and the user can switch to the "Document View". In the DD, documents are queued and can be opened by the user on a simple tap. Documents open in standard A4 format, just like paper sheets for ease of reading. The user can drill down one by one to decide which documents are relevant to the search, and which ones are non-relevant. Touching the "relevant" tab on the order of a document will tag that document and move it to a container called the "relevant bucket" (at the bottom right of the touchscreen); and touching the "non-relevant" tab will do the same but to the "non-relevant bucket" (at the bottom left corner).

To identity and locate potentially interesting data, the user can manipulate specific search widgets called "Virtual Magnets" (VM): a VM needs first to be populated with a term chosen by the user; then the user can move the magnet widget close to a group of documents (e.g. a cluster), which pulls out all the documents holding the chosen term. The tiles representing these documents are attracted around the magnet which helps users quickly visualize how many documents meet their search criteria, (see Figure 1). A swipe on the group of tiles gathered around the magnet automatically opens a random sample of documents, that the user can read to further decide if the subset is worth inspecting. To further review the subset, the user moves the document subset, from the magnet location to the Document Dispenser, through a 2 hand gesture.

Magnet Widgets can be populated in three different ways: a) *Static keywords:* a tap on a magnet opens a wheel menu displaying user-predefined terms; a tap on a term closes the magnet menu and populates the VM with the chosen term that appears on top of the widget; b) *Extracted keywords:* user chooses among keywords automatically extracted from each cluster by the clustering algorithm (or named entities) and displayed on the touchscreen; by touching with one finger a term listed to the right side of a cluster, and subsequently touching with the other hand a VM widget, the user can see the chosen term navigating to the magnet widget, and finally appearing displayed on top of the widget; c) *Highlighted keywords:* the user can directly highlight some text segments with his/her finger from a document displayed in paper format on the screen; when the user's finger is released from the document, a magnet pops-up with the selected text appearing on top of the widget.

In Disco the user switches easily between iterative/lookup search and exploratory search: an Iterative Search will correspond to a systematic drill down on documents stacked in the DD, by opening, reading and tagging them to the relevant or non-relevant bucket. An Exploratory Search will correspond to an expansion of the search to new areas of the document wall or groups of data, using clustering, categorization, or term-based filtering via VMs. At any time, users can interrupt an iterative search, and switch to an exploration phase. This typically happens as the review session unfolds and documents are read and labelled: knowledge is acquired and new information is discovered; interest drifts occur that can lead to new exploration phases. In this work we focus on using semantic relatedness in order to increase the level of exploration of the data in a simple and intuitive way.

### 2.2 Generating a Model for Semantic Relatedness

The notion of "Semantic Relatedness" is meant to quantify the semantic relationship between words, sentences or concepts, in a broad sense, covering relations beyond similarity such as: *"is-a-kind-of", "is-a-part-of", "is-a-specific-example-of", "is-the-opposite-of", etc.* Word embeddings are used to build semantic language models that can be afterwards deployed to obtain the semantic information on input terms: either getting the level of relatedness between 2 input words (or phrases), or finding lists of most semantically related terms given an input word.

We applied Semantic Relatedness for semantic search. It is important here to make a distinction with the notion of Semantic Web (SW): the SW, (which has been around at least since 2001) is a web of linked data with a semantic structuring achieved by ontologies and supported by several technologies and standards (i.e. RDF, OWL, SparQL, etc). A number of SW search engines are available, such as Siren built on top of Lucene$^{\text{TM}}$/Solr[1].

In contrast, when building a semantic model through word embeddings, the attempt is to learn the

[1] https://github.com/sirensolutions/siren

| Datasets | Pearson | Spearman |
|---|---|---|
| ALL | 0.65045 | 0.6699 |
| MC30 | 0.7904 | 0.7835 |
| RG65 | 0.7614 | 0.7626 |
| MTurk | 0.7020 | 0.6738 |
| WordSim353-Sim | 0.6696 | 0.7183 |
| WordSim353-Rel | 0.5147 | 0.5386 |

**Table 1:** Model evaluation on different datasets.

context of words in an unsupervised way from unstructured raw corpora. In this work, we used Google's word2vec toolkit[2], see (Mikolov et al., 2013a; Mikolov et al., 2013b; Mikolov et al., 2013c). As a large dataset is required to build a model generic enough to serve in different domains, we collected a large set of data (approximately 40GB) using the following sources:

- The training monolingual news crawl in 2012 & 2013 of the 9th Workshop on SMT.[3]
- The 1-billion-word language model benchmark.[4]
- The UMBC webbase corpus.[5]
- The latest Wikipedia dump file.[6]

To integrate data of heterogeneous format, we applied some pre-processing: firstly, converting all texts to lower case and removing special characters; secondly for the Wikipedia data, keeping only the body text in <text> tags, (removing xhtml tags, image links, etc) to get a dataset of 28GB.

We generated the semantic model via the Google's word2vec toolkit using the combination of Skip-Gram and Negative sampling as recommended as the best strategy (Mikolov et al., 2013b). In addition, we used the "word2phrase" function to get a model supporting also n-grams. Finally, we obtained a model of 4.4GB.

For model evaluation, as we wanted to evaluate the model capability of finding semantically related words to be used in our semantic search, we tested the model on the task of computing the semantic similarity/relatedness between words. We build the evaluation data from several datasets: MC30 (Miller and Charles, 1991), RG65 (Rubenstein and Goodenough, 1965), MTurk (Radinsky et al., 2011), WordSim353 Similarity and Relatedness (Agirre et al., 2009). It contained 837 word pairs in total with human annotation for semantic similarity and relatedness. However, since these datasets were developed and annotated by different people and annotation guidelines, the semantic similarity/relatedness scores were specified in different scales; thus we normalized annotation score via Feature Scaling[7] to [0-1]. We used the Pearson[8] and Spearman[9] correlation methods for evaluation metrics. Table 1 shows the results of our model evaluation on different settings of datasets. It shows that our semantic model obtains good results on several datasets compared to other models reported on this site[10]. The semantic model is further used in Disco to assist users during search sessions on collections of text documents. The next section explains how the model is loaded and queried by the searchers.

### 2.3 Loading and Querying the Model in Disco

Semantic relatedness capabilities are provided by the "Disco Semantics" Java library, that can: a) load a model in memory; b) query the model from an input term, to get a list of most related words/phrases; c) compute the semantic relatedness score between two words. The model is loaded at application start-up to ensure users can access it in real-time during a search session. Loading can take a few minutes (e.g.

[2] https://code.google.com/archive/p/word2vec

[3] http://www.statmt.org/wmt14/translation-task.html

[4] http://www.statmt.org/lm-benchmark/1-billion-word-language-modeling-benchmark-r13output.tar.gz

[5] http://ebiquity.umbc.edu/redirect/to/resource/id/351/UMBC-webbase-corpus

[6] https://en.wikipedia.org/wiki/Wikipedia:Database_download

[7] https://en.wikipedia.org/wiki/Feature_scaling

[8] https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient

[9] https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

[10] http://aclweb.org/aclwiki/index.php?title=Similarity_(State_of_the_art)

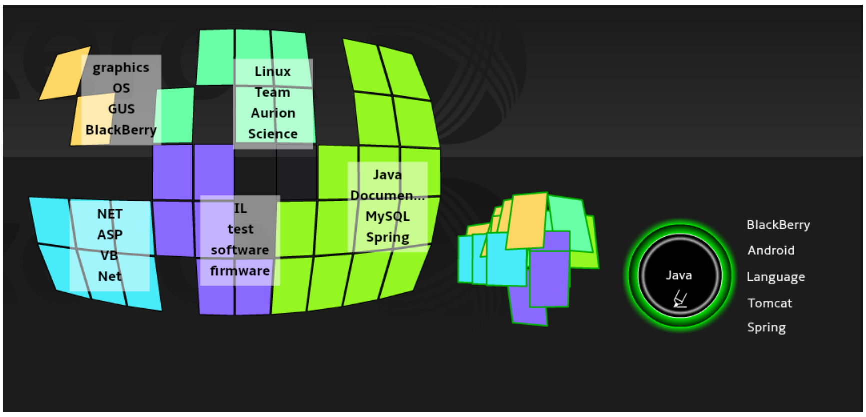

**Figure 1:** Result of a magnet query in semantic mode from the term "Java".

at the worst case, $\approx 6mn$ for the 4.4GB model on an ordinary computer - 8Gb ram); then computing the relatedness score for a word-pair is fast (less than 1s, again on an ordinary machine).

Online requests to the semantic module are made using the virtual magnet widgets: the user selects a word, phrase or text fragment as input to populate a magnet, that can operate in a "semantic relatedness mode". The searcher can select the input from 3 different sources: *Static keywords* (user predefined terms via magnet menus); *Extracted keywords* (discovered by the system via clustering/entity extraction, and displayed on the screen); *Highlighted keywords* (directly selected from document contents by the user). The reuse of existing text through natural hand gesture for formulating queries on the TUI is particularly convenient, and it facilitates exploratory search behaviour by enabling sequential search (see section 2.4).

Once the magnet is populated in semantic mode, the system computes on-the-fly the list of semantically related terms to form an expanded query. An animated glow effect on the widget indicates that it is ready for searching for new documents. When moved close to a group of documents, the magnet attracts all documents that match one or several of the terms from the expanded query, (see Figure 1). The searcher can choose to further inspect the retrieved documents by sending them to the Document Dispenser for a systematic review. The magnet can also be applied to other groups of documents to locate other sources of information in the data space.

## 2.4 Visualizing Related Terms and Formulating Subsequent Queries

On the touch screen, the list of "semantically related words" is displayed next to the magnet that operated the query, so that the searcher can instantly visualize and access them: users can scroll and select items, each item showing a related word. The displayed items are ranked by distance, i.e. the top item is the one most similar to the input word populating the magnet, etc. Whenever the user drags the magnet to another location on the screen, the list stays close to the magnet, following its movement. The maximum number of "related words" displayed and used during a query is defaulted to 10 which can be configured (upfront or on the fly). As the items displayed in the list of semantically related terms are also selectable, they can be used in turn for populating a new magnet, then launching a new query through 10 other semantically related terms computed on the fly, and so on, enabling in this way to run sequential semantic searches.

## 2.5 Implementation

Disco is based on a client-server architecture: the client TUI is developed in Adobe Air; the server orchestrating the text mining components (ML categorization, clustering, rule-based entity extraction, semantic search) is implemented in Java. The Disco Semantics component uses Deeplearning4J (http://deeplearning4j.org/) which is a Java open-source deep-learning library distributed under Apache license. We use it for querying the semantic model, whereas the model itself is off-line generated using word2vec.

## 3 Conclusion and Future Work

Technology-Assisted Review tools find applications in various domains and can be embedded in a range of industrial services, (e.g. in eDiscovery). With Disco, we combine semantic search and a specific design approach on a TUI, to increase information discovery on collections of textual documents. We aim at making semantic relatedness techniques available to all - and especially nontechnical-users, in a simple, generic and effective way. User studies show that a specific system design associated with touch capabilities can lead to more active search behaviours, (in addition to shortening system learning curve and allowing for faster adoption). We plan now to set-up a user study to collect feedback on the usability of the information provided by the semantic model, and the interaction design built to leverage semantic relatedness in search sessions.

## References

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.

Khalil Klouche, Tuukka Ruotsalo, Diogo Cabral, Salvatore Andolina, Andrea Bellucci, and Giulio Jacucci. 2015. Designing for exploratory search on touch devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 4189–4198. ACM.

Gary Marchionini. 2006. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, pages 746–751.

George A Miller and Walter G Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Caroline Privault, Jacki O'Neill, Victor Ciriza, and Jean-Michel Renders. 2010. A new tangible user interface for machine learning document review. *Artificial Intelligence and Law*, 18(4):459–479.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM.

Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Barbara M Wildemuth and Luanne Freund. 2012. Assigning search tasks designed to elicit exploratory search behaviors. In *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*, page 4. ACM.

Xerox. 2010. Inside innovation at xerox: Smart document review technology. `https://www.youtube.com/watch?v=ZwPU51j5qoU`.