Contextual Insights

Ariel Fuxman[†], Patrick Pantel[†], Yuanhua Lv[†], Ashok Chandra[†], Pradeep Chilakamarri[†],
Michael Gamon[†], David Hamilton[†] Bernhard Kohlmeier[†], Dhyanesh Narayanan[†],
Evangelos Papalexakis[‡*], Bo Zhao[†]
{arielf, ppantel, yuanhual, achandra, mgamon, dahami, dhyann, bozha}@microsoft.com
{pradeep.chilakamarri, bernhard.kohlmeier}@microsoft.com, epapalex@cs.cmu.edu
[†] Microsoft   [‡] Carnegie Mellon University

ABSTRACT

In today's productivity environment, users are constantly researching topics while consuming or authoring content in applications such as e-readers, word processors, presentation programs, or social networks. However, none of these applications sufficiently enable users to do their research directly within the application. In fact, users typically have to switch to a browser and write a query on a search engine. Switching to a search engine is distracting and hurts productivity. Furthermore, the main problem is that the search engine is not aware of important user context such as the book that they are reading or the document they are authoring. To tackle this problem, we introduce the notion of contextual insights: providing users with information that is contextually relevant to the content that they are consuming or authoring. We then present Leibniz, a system that provides a solution for the contextual insights problem.

1. INTRODUCTION

We introduce the notion of *contextual insights*: providing users with information ("insights") that is contextually relevant to the content that they are consuming or authoring. As an example, consider a user who is reading an article on President Obama's address to the nation on the Syrian crisis. At some point, the user may highlight the term "Russia" and ask the system for contextual insights. Figure 1 shows a screenshot of the resulting contextual insights experience. These results include the Wikipedia articles "Russia", "Russia's role in the Syrian civil war" and "Russia-Syria relations". Now, suppose that the user highlights a different term: "weapons". Then, a good contextual insights result might be the Wikipedia article "Syria and weapons of mass destruction". Clearly, these results are dependent on the context of the document that the user is reading.

At a high level, the input to the contextual insights problem consists of: 1) A *focus of attention*, 2) a *context*, and 3) a

---
[*]Work carried out as an intern at Microsoft Research.

Figure 1: Contextual Insights experience.

knowledge base. The output consists of elements from the knowledge base (Web pages, entities, etc.) that are relevant to the focus of attention and the context. In this paper, we present a system, called Leibniz, that provides a solution for the contextual insights problem, implementing a general architecture that leverages existing search engine technology; and an experimental evaluation of the Leibniz contextual insights system.

Contextual insights is related to *entity linking* [1]: In our example, entity linking techniques can readily produce the Wikipedia article on the country "Russia" (as opposed to, say, the town called "Russia" in Ohio), but obtaining results that are contextually relevant to the document, such as "Russia's role in the Syrian civil war" is beyond their scope. It is also related to previous work on web search within context [2, 3]. Like conventional web search, existing contextual search systems assume that (1) users explicitly provide the queries that they have in mind; and (2) a bad set of results is worth the same as no result at all. Both assumptions are dropped in contextual insights, which introduces the need for components such as prediction of focus of intent and context-based filtering of results.

2. CONTEXTUAL INSIGHTS FRAMEWORK

The architecture of the Leibniz system is shown in Figure 2. Its main components are *smart selection*, which predicts the focus of attention given some text highlighted by the user; *context extraction*, which chooses terms from the context to be used to build an appropriate query; *query formulation*, which builds the actual query for the search engine; and *result post-processing*, which re-ranks and filters the results of the search engine in a context-aware fashion.
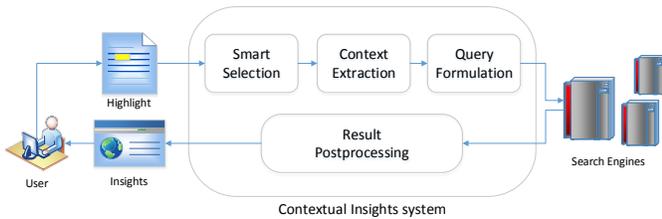
Figure 2: The Contextual Insights architecture.

To illustrate the architecture, suppose that a user is reading the article on the Syrian crisis on a touch-enabled device, and taps on the word "Federation" in the sentence "The Russian Federation has proposed a plan for the destruction of Syria's chemical weapons". The smart selection component would then predict that even though the user highlighted "Federation", the intended focus of attention is "Russian Federation". The context extraction component would choose representative context terms such as "Syria" and "chemical weapons". The query formulation component would then build a query such as `russian federation (syria or "chemical weapons")`, and issue it to the search engine. Finally, the post-processing component would use the entire context of the focus and the output of the search engine to decide which results are to be shown as insights.

Some key design choices include:

- *Predict the focus of attention.* In our example, it is unlikely that sending the query `federation` to the search engine will produce results related to Russia. Thus, we need to predict the focus of attention in order to form a meaningful query.
- *Add context terms to the query.* One might imagine a different design option that would simply send the terms of the focus of attention to the search engine. The problem with such approach is that search engines return only a limited number of results. So if we just send the query `russian federation` to the search engine, it is unlikely that that it will return any result related to the Syrian crisis context.
- *Adapt the results of the search engine via post-processing.* Conventional information retrieval systems always return results if the corpus contains documents lexically related to the query. In contrast, in the contextual insights problem, it is acceptable, and occasionally even desirable, not to show any insights for a given focus.

3. EXPERIMENTAL EVALUATION

3.1 Experimental Setup

We employed a corpus consisting of all English textbooks from the Wikibooks site. We randomly sampled 100 books from the corpus, and one paragraph from each book. The paragraphs were then shown to a human annotator, using a UI that shows the paragraph in the context of the page of the book where it appears. The annotator was asked to choose from the paragraph any phrases for which she would like to see additional information using an external resource (such as an encyclopedia, Wikipedia, a search engine, etc.) This resulted in a total of 337 selected phrases, which constitutes the set of focus phrases that we consider in the experiments.

For each experiment, we ran every focus in the test set through the system, which resulted in a set of (focus, insight) pairs. The annotation was done by an independent vendor, who was provided with the appropriate guidelines and a user interface designed for that purpose. We measured the system using *precision* and *recall*.

3.2 Results

The goal of these experiments is to evaluate the end-to-end quality of Leibniz as a contextual insights system. To do so, we compare the following two systems:

- The Leibniz contextual insights system, where we consider 100 words to each side of the focus as context. All parameters are tuned based on a validation set We use Wikipedia as knowledge base (i.e., we add `site:en.wikipedia.org` to the generated queries).
- A baseline system that consists of simply sending the focus to a search engine, restricted to Wikipedia.

Figure 3 shows precision and recall at positions 1 and 3. For Leibniz, we plot a precision and recall curve corresponding to different threshold values from the post-processing component. There is a single precision-recall value for the baseline since it is not parameterized (i.e., it consists of a simple call to the search engine). We can observe that Leibniz consistently outperforms the baseline.
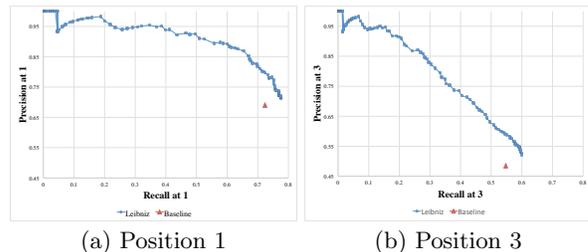

(a) Position 1   (b) Position 3

Figure 3: End-to-end precision-recall

4. CONCLUSION

This paper introduces the concept of *contextual insights*, where a user seeks insightful results for a certain word or phrase, taking the context into account. We presented a system called Leibniz that implements a solution for the contextual insights problem and evaluated the relevance of its end-to-end results.

5. REFERENCES

[1] S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. In *EMNLP-CoNLL*, pages 708–716, June 2007.

[2] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: The concept revisited. In *WWW*, pages 406–414, 2001.

[3] R. Kraft, C. C. Chang, F. Maghoul, and R. Kumar. Searching with context. In *WWW*, pages 477–486, 2006.