A tool for extracting sense-disambiguated example sentences through user feedback

Beto Boullosa[†] and **Richard Eckart de Castilho**[†]
and **Alexander Geyken**[‡] and **Lothar Lemnitzer**[‡]
and **Iryna Gurevych**[†]

[†]**Ubiquitous Knowledge Processing Lab**
Department of Computer Science
Technische Universität Darmstadt
http://www.ukp.tu-darmstadt.de

[‡]**Berlin-Brandenburg Academy of Sciences**
http://www.bbaw.de

Abstract

This paper describes an application system aimed to help lexicographers in the extraction of example sentences for a given headword based on its different senses. The tool uses classification and clustering methods and incorporates user feedback to refine its results.

1 Introduction

Language is subject to constant evolution and change. Hence, lexicographers are always several steps behind the current state of language in discovering new words, new senses of existing words, cataloging them, and illustrating them using good example sentences. To facilitate this work, lexicographers increasingly rely on automatic approaches that allow sifting efficiently through the ever growing body of digitally available text, something that has brought important gains, including time saving and better utilizing limited financial and personal resources.

Among the tasks that benefit from the increasing automation in lexicography is the automatic extraction of suitable corpus-based example sentences for the headwords in the dictionary. Our paper describes an innovative system that handles this task by incorporating user feedback to a computer-driven process of sentence extraction based on a combination of unsupervised and supervised machine learning, in contrast to current approaches that do not include user feedback. Our tool allows querying sentences containing a specific lemma, clustering these sentences by topical similarity to initialize a sense classifier, and interactively refining the sense assignments, continually updating the classifier in the background.

In the next section, we contextualize the task of example extraction; section 3 describes our system; section 4 is devoted to evaluation; section 5 summarizes the conclusions and future work.

2 Extraction of Dictionary Examples

Example sentences can help understanding the nuances of the usage of a certain term, specially in the presence of polysemy. This has become rather important in the last decades, with the shift that has occurred, in the field of dictionary making, from a content-centered to a user-centered perspective (Lew, 2015). With the popularization of online dictionaries, space-saving considerations have lost the importance once held, making it easier to add example sentences to a given headword.

Didakowski et al. (2012) argue that a system for example extraction should ideally act "like a lexicographer", i.e., it should fully understand the examples themselves, something arguably beyond the scope of current NLP technology. Instead, operational criteria must be used to define "good" examples, as seen also in the work of Kilgarriff et al. (2008), criteria like presence of typical collocations of the target word, characteristics of the sentence itself, and guaranteeing that all senses of the target word are represented by the extracted example sentences.

Several methods to automate the task have been developed, the most popular being GDEX ("Good Dictionary EXamples") (Kilgarriff et al., 2008). GDEX is a rule based software tool that suggests "good" corpus examples to the lexicographer according to predefined criteria, including sentence length, word frequency and the presence/absence of pronouns or named entities. The goal of GDEX is to reduce the number of corpus examples to be inspected by extracting only the n-"best" examples, the default being 100 sentences. It has been used and adapted for languages other than English.

Didakowski et al. (2012) presented an extrac-

Proceedings of the EACL 2017 Software Demonstrations, Valencia, Spain, April 3-7 2017, pages 69–72

tor of good examples based on a hand-written, rule-based grammar, determining the quality of a sentence according to its grammatical structure combined with some simpler features as used by GDEX. None of those works, however, focused on differentiating example sentences according to the word senses or the target words.

Cook et al. (2013) used Word Sense Induction (WSI) to identify novel senses for words in a dictionary. They utilized hierarchical LDA (Latent Dirichlet Allocation) (Teh et al., 2006), a variation of the original LDA (Blei et al., 2003) topic model, to identify novel word senses, later combining this approach with GDEX, to allow extracting good example sentences according to word senses (Cook et al., 2014). However, they obtained "encouraging rather than conclusive" results, specially due to limitations of the LDA approach in linking identified topics with word senses.

Our work explores and develops a similar approach of using topic modeling and WSI to cluster sentences according to the senses of a target word, but we take a step further, using the initial clusters as seed for a series of interactive classification steps. The training data for each classification step are sentences whose confidence scores calculated by the system exceed a threshold, and sentences manually labeled by the user. The process leads to a user-driven refinement in the labeling process.

3 System description

The computer-assisted, interactive sense disambiguation process supported by our system involves: 1) import sentences into a searchable index; 2) retrieve sentences containing a specific word (lemma); 3) cluster selected sentences, providing a starting point for interactive classification; 4) train a multi-class classifier from the initial clusters - that trains on the most representative sentences for each cluster and is then used to label the rest of the sentences; a sentence is "representative" if the confidence score calculated by the system exceeds a configurable threshold; 5) refine the classifier by interactively correcting sense assignments for selected sentences.

The system supports multiple users working in so called **projects**, which define, among other things, the list of stopwords available for clustering and classification and the location where the sentences should be retrieved from.

A project contains **jobs**, corresponding to tasks performed over a certain headword (actually defined by its lemma and POS tag). Tasks include searching for initial sentences, clustering and classification. Users can work on many jobs in parallel and isolated, which allows calculating inter-rater agreement on the sense disambiguation task.

As for the technologies, the tool was developed using Java Wicket and relying on Solr[1], Mallet [2], DKPro Core (Eckart de Castilho and Gurevych, 2014), Tomcat and MySQL. The next subsections describe the system in more detail.

3.1 Searching

After starting a job, the user goes to the Search page to look for sentences containing the desired lemma. They are shown in a KWIC (Keyword in Context) table with their ids. When satisfied with the results, the user selects the stopwords to use in the next steps and goes to the Clustering phase.

3.2 Clustering

In the clustering page, the selected sentences are automatically divided into clusters corresponding to topics that ideally relate to the senses of the target word. The user manually configures how many clusters the topic modeling generates and control the hyper-parameters to fine-tune the process. We currently use Mallet's LDA implementation to topic modeling (McCallum, 2002).

The clustering page lists the selected sentences according to the generated clusters. Each cluster is shown in its own column, with a word cloud on the top, containing the main words related to it, which helps the user to assess cluster's quality and meaning. The word sizes correspond to their LDA-calculated weights. The user can change hyper-parameters and regenerate clusters as often as desired, before proceeding to the classification step.

3.3 Classification

In the classification step, the user interactively refines the results, giving feedback to the automatic classification in order to improve labeling of the sentences according to word senses. The initial automatic labels correspond to the results of the clustering phase. The user starts analyzing each sentence and decides if the automatically assigned sense label is appropriate or if a different label needs to be assigned manually. The initial default

[1] http://lucene.apache.org/solr/

[2] http://mallet.cs.umass.edu

Figure 1: Classification page

User	Assist	Accuracy	Time (mm:ss)
Annot. 1	No	0.96	8:05
Annot. 1	Yes	0.95	6:05
Annot. 2	No	0.90	10:05
Annot. 2	Yes	0.90	7:55

Table 1: Evaluation results

value for a sentence's manual label is **"unseen"**, indicating that the user has still not evaluated that sentence. Besides the available sense labels, two special labels can be assigned to a sentence: **"neither"**, to indicate that none of the available labels is applicable; **"unknown"**, meaning that the user does not know how to label it.

The classification page (figure 1 - the numbers below correspond to its elements) has a table listing each sentence with its id ①; automatically assigned sense labels in the previous ⑤ and current ⑥ iterations; confidence scores (weights) of the sense label in the previous ③ and current ④ iterations; manually assigned sense label ⑦; sentence text ⑧; and an indicator to tell if the sense label has changed between the previous and current iteration ②. The page has also a widget detailing the different word senses currently available ⑩. Besides editing the label of a sense, the user can also add new manual senses ⑪.

After modifying parameters, adding senses and manually labeling sentences, a new classification iteration can be started. The classifier uses the threshold ⑨ to identify the training data - the confidence scores are calculated by the classifier (although in the initial classification they come from the topic modeling). Furthermore, manually labeled sentences are also used as training data. We use the Naive Bayes algorithm, a classical approach for Word Sense Disambiguation, known for its efficiency (Hristea, 2013). We use the Mallet implementation of Naive Bayes, with the sentence tokens as features.

4 Evaluation

To evaluate the tool, we conducted experiments in two different scenarios: 1) using no assistive features, annotators classified sentences identifying the word senses by their own; 2) using automatically-generated clusters, annotators let the system suggest senses and then manually assigned labels to sentences, helped by the feedback-based multi-class classifier. Every annotator applied the two scenarios to a target word, namely "Galicia"[3], with three senses: a) the Spanish autonomous community, b) the region in Central-Eastern Europe; c) a football club in Brazil.

The senses were randomly distributed over 97 sentences in the first scenario (38/46/13 sentences for the respective senses) and 99 in the second (43/41/15). Sentences in both scenarios did not overlap, and were taken from a larger dataset of manually annotated sentences. The experiments were performed by two non-lexicographers (computer scientists with NLP background). We measured the time taken in every scenario and calculated the accuracy of the final results compared to the manually annotated gold standard.

Results (table 1) indicated that the time was significantly reduced when working with full assistance, compared to working without assistance. Although accuracy did vary very little, there was

[3]Although proper nouns are not present in conventional dictionaries, but rather in onomastic dictionaries, which usually do not make use of examples, it serves well, for methodological reasons, to our evaluation purposes.

a slight loss of quality in the results of the first annotator when using assistance. This might indicate a negative influence of system suggestions on the annotator or it could be attributable to a more difficult random selection of the samples in certain cases. These effects call for further investigation. However, using the tool outside the evaluation setup, we noted subjective speedups from developing smart strategies to optimize the use of the information provided by the machine (e.g. quickly annotating sentences containing specific context words or sorting by sense and confidence score). Thus, we expect the automatic assistance to have a larger impact in actual use than the present evaluation can show.

5 Conclusions and future work

We have introduced a novel system for interactive word sense disambiguation in the context of example sentences extraction. Using a combination of unsupervised and supervised methods, corpus processing and user feedback, our approach aims at helping lexicographers to properly assess and refine a list of pre-analyzed example sentences according to the senses of a target word, alleviating the burden of doing this manually. Using various state-of-the-art techniques, including clustering and classification methods for word sense induction, and incorporating user feedback into the process of shaping word senses and associating sentences to them, the tool can be a valuable addition to a lexicographer's toolset. As next steps, we plan to focus on the extraction of "good" examples, adding support for ranking sentences in different sense clusters according to operational criteria like in Didakowski et al. (2012). We also plan to extend the evaluation and to observe the strategies that users develop, in order to discover if they can inform further improvements to the system.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme (H2020-EINFRA-2014-2) under grant agreement No. 654021. It reflects only the author's views and the EU is not liable for any use that may be made of the information contained therein. It was further supported by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 01UG1416B (CEDIFOR) and by the German Research Foundation under grant No. EC 503/1-1 and GU 798/21-1 (INCEpTION).

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March.

Paul Cook, Jey Han Lau, Michael Rundell, Diana McCarthy, and Timothy Baldwin. 2013. A lexicographic appraisal of an automatic approach for detecting new word-senses. In *Proceedings of eLex 2013*, pages 49–65, Tallinn, Estonia.

Paul Cook, Michael Rundell, Jey Han Lau, and Timothy Baldwin. 2014. Applying a word-sense induction system to the automatic extraction of diverse dictionary examples. In *Proceedings of the XVI EURALEX International Congress*, pages 319–328, Bolzano, Italy.

Jörg Didakowski, Lothar Lemnitzer, and Alexander Geyken. 2012. Automatic example sentence extraction for a contemporary german dictionary. In *Proceedings of the XV EURALEX International Congress*, pages 343–349, Oslo, Norway.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11, Dublin, Ireland, August.

Florentina T. Hristea. 2013. The naïve bayes model in the context of word sense disambiguation. In *The Naïve Bayes Model for Unsupervised Word Sense Disambiguation: Aspects Concerning Feature Selection*, pages 9–16. Springer, Heidelberg, Germany.

Adam Kilgarriff, Milo Husk, Katy McAdam, Michael Rundell, and Pavel Rychlý. 2008. Gdex: Automatically finding good dictionary examples in a corpus. In *Proceedings of the XIII EURALEX International Congress*, pages 425–432, Barcelona, Spain.

Robert Lew. 2015. Dictionaries and their users. In *International Handbook of Modern Lexis and Lexicography*. Springer, Heidelberg, Germany.

Andrew Kachites McCallum. 2002. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.