Alexandru Ioan Cuza University

Faculty of Computer Science



Semantic Similitude and Text Classification

**DOCUMENTATION**

**Authors:** A3 Student Group

**Coordinator:** Ionuţ Pistol

2016

**ABSTRACT**

In this paper we describe the work carried out in order to implement a system which allows users to define and discover semantic categories for short texts. A label (category) can be defined either aș a set of lexical features or a set of texts exemplifying that label. Texts can also be clustered around included lexical elements, which may help discover and define new labels. The user can use the provided UI to manage the semantic labels and texts processed by the system. The main difficulties faced in developing the system were mainly regarding defining a semantic label, processing the text and associating a text with accurate labels.

**Author Keywords**

Semantic categories; text processing; human-computer interfaces

**ACM Classification Keywords**

I.2.7 Artificial Intelligence. Natural Language Processing

**INTRODUCTION**

Associating semantic labels to texts is one of the most relevant Lantural Language Processing tasks, aș it is one of the first steps required în order to manage the large (and increasing) quantities of text available. For the average user, the ability to sort, filter and group e-mails, news, forum posts, can have significant benefits în time saved accessing data and in minimising the risk of missing important information. Describing the semantic context of a text has been one of the main priorities of NLP researchers [1], [2]., solutions proposed offering significant insights into how humans understand the meaning of a text but failing to offer a universal method of transforming that meaning into a universally accepted computer-friendly representation. Many proposed solutions [3] [4] [5] offer pragmatic approaches to the problem, either by simplifying the requirements to matching text to semantic representations, finding particular data (like in Question Answering systems) or assigning semantic labels to texts.

Section 2 of this paper briefly describes the way semantic labels are managed in our system, section 3 showing how those labels are automatically associated to short texts. Section 4 indicates the consideration made while designing an UI and shows how the implemented functionalities can be used in the current UI. Section 5 is a brief overview of the current state of the system and the planned further development efforts.

**SEMANTIC LABELS**

Managing semantic labels in our system was done using Apache OpenNLP. The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution. These tasks are usually required to build more advanced text processing services.

That way, the keywords and phrases are analyzed by the application and assigned to a category that has been defined in the same way. Simply put, a text file called model.train is created with a new, unnamed category along with the keyword or phrase.

After adding a new characteristic to a certain category, whether it is a keyword or a phrase (specified in a single row), the application must be executed again to be trained and to update that specific model (category).

Every time we execute the application, the model, which is a file called model.bin, is being updated.

**TEXT LABELLING**

The first problem faced when trying to apply machine learning in a practical setting is selecting attributes for the data at hand. This is probably also the most important problem: if you don’t choose meaningful attributes—attributes which together convey sufficient information to make learning tractable—any attempt to apply machine learning techniques is doomed to fail. In truth, the choice of a learning scheme is usually far less important than coming up with a suitable set of attributes. In our application, we use rather simplistic attributes: they count the number of times specific keywords appear in the message to be classified. We assume that each message is stored in an individual file, and the program is called every time a new message is to be processed. If the user provides a class label for the message, the system will use the message for training; if not, it will try to classify it.

**LABELME SYSTEM**

In our application, the main page will have three buttons: add texts, add category and email. All of these three buttons are placed to the right side of our application for a better view.

After you press on add texts, the application will redirect us to another page where we can search and add new texts from our computer (with Browse).

When we press on add category button, we will be able to create a new category of texts. We need to choose a specific word for our category. After this, we must write a list with keywords (separated by commas) or just one keyword.

Finally we have the email button where the user allows application to connect to his email for retrieve the messages.

**CONCLUSIONS**

LabelMe is a cross-platform application used to categorize short texts or e-mail messages. It lets the user load texts via “.txt” files or connect with an existing Gmail account, so that the application can access and categorize existing e-mails.

Messages will be processed and labeled into an existing category, if any, using a special algorithm that checks for specific keywords. The user can also define his own categories by providing a few keywords that define the custom category.

The current state of the application can be improved by adding the possibility to connect with other e-mail providers such as Yahoo, Hotmail etc. and by making the categorizing algorithm to deliver higher accuracy results.

**ACKNOWLEDGMENTS**

The development work described in this paper was carried out by members of the A3 student group at the Faculty Of Computer Science, “Alexandru Ioan Cuza” University of Iași.

**REFERENCES**

1. Van Dijk, Teun Adrianus. "Text and context explorations in the semantics and pragmatics of discourse." (1980).

2. Fillmore, Charles J. "Lexical semantics and text semantics." New directions in linguistics and semiotics 123 (1984): 147.

3. Aas, Kjersti, and Line Eikvil. "Text categorisation: A survey." (1999).

4. Martínez-Torres, María del Rocío, et al. "A text categorisation tool for open source communities based on semantic analysis." Behaviour & Information Technology 32.6 (2013): 532-544.

5. Šajgalík, Márius, Michal Barla, and Mária Bieliková. "Exploring multidimensional continuous feature space to extract relevant words."Statistical Language and Speech Processing. Springer International Publishing, 2014. 159-170.