A System For Large Scale Data Exploration and Organization

Luis Rei, Blaz Fortuna, Marko Groblenik, and Dunja Mladenić
Artificial Intelligence Laboratory, Jožef Stefan Institute and Jožef Stefan International Postgraduate School
Jamova cesta 39, 1000 Ljubljana, Slovenia
Tel: + 386 14773528; fax: + 386 14773851
e-mail: {luis.rei, blaz.fortuna, marko.grobelnik, dunja.mladenic}@ijs.si

ABSTRACT

We present a semi-automatic data exploration and organization tool. The system integrates machine learning and text mining algorithms into an simple user interface and a Client/Server architecture. The main features of the systems include unsupervised and supervised methods for concept suggestion, visualization and ability to make both data and methods available to other applications as a service.

1 INTRODUCTION

This paper presents an approach and system intended to support the user in managing textual information overload and expert scarcity by leveraging ontologies and machine learning to semi-automatically organize information. The developed system is focused on the ease of use and simple integration of discovered models with other software. It can interactively handle large datasets in the gigabyte range. Small modifications to its interactive interface could easily allow it to handle datasets over one terabyte.

The tool is meant for use by anyone who wants to rapidly understand unlabeled data and/or build classifiers and services from it. Regardless of whether they are machine learning experts, domain experts or not. Some of the methods used in the proposed tool were first proposed, explored and evaluated in OntoGen [1, 2], a desktop application built for semi-automatic construction of topic ontologies from text corpora.

We will begin by providing an overview of the architecture, its relative advantages. Following, we describe some of the methods and algorithms used. Next, we detail an example use case. Lastly we will present our planned future work.

2 ARCHITECTURE OVERVIEW

The developed system, Elycite[1] , is an Open Source Client/Server application. We built it on top of QMiner[2] , an open source data analytics platform for processing large-scale real-time streams of structured and unstructured data. QMiner is used as a database and server where the data is stored and where all the machine learning algorithms are executed. The current Elycite version works with textual data, however QMiner allows the user to load any data that can be represented as a series of JSON records. Elycite allows the user to choose which field to use for different steps in the exploration process. The client component of the software is a browser based interface. This Client/Server architecture provides several important characteristics:

- **Computational capability** - the increase in the amount of data to be analyzed requires an increase computational capability. The previously mentioned OntoGen focused on smaller datasets which could be handled by a personal computer. Elycite focus on much larger datasets while allowing it's interface to run on portable computers and tablets.

- **Collaborative** - the increase in interest of data analysis and large amounts of data to analyze, in some cases, means that more people need to work together to sift through it. With Elycite, several people can work together, possibly simultaneously, to organize the same data, and build an ontology and classifiers collaboratively by sharing the same server, while using different clients.

- **Services** - the server component provides REST [3] services which can be used by either the client component or other applications. For example, a concept can be easilty changed into a classifier and exposed through a web service. Other applications can use this web service to classify documents into the developed ontology without the need to implement any additional steps.

- **Web Interface** - Web interfaces became popular during the early 2000s. A lot of traditional desktop software has or is moving to the Web. The adoption of smartphone and tablets has left the Web as the common interface across all platforms. Our tool provides a simple, interface, with multiple visualizations available and hidden-by-default *Advanced options*.

[1] Elycite: https://github.com/lrei/elycite
[2] QMiner: http://qminer.ijs.si/

3 METHODS

Elycite works on documents, more precisely, QMiner records. Each individual document is known as an instance. The user organizes the documents into sets of related documents which form concepts. Concepts can have sub-concepts. For example, we can think of all news articles that are related to sports and group them together under the concept *Sports*. If we then group all articles related to football and call it the concept *Football*, we naturally we consider the latter to be a sub-concept of the former. The concepts and the relationships between them together form an Ontology. Concepts can be created manually or with the aid of machine learning techniques. We have combined a set of unsupervised, semi-supervised and supervised techniques for organizing and exploring a dataset. All algorithms work on a common feature representation. The features are extracted using preprocessing techniques for textual data such as stemming and stop-word removal.

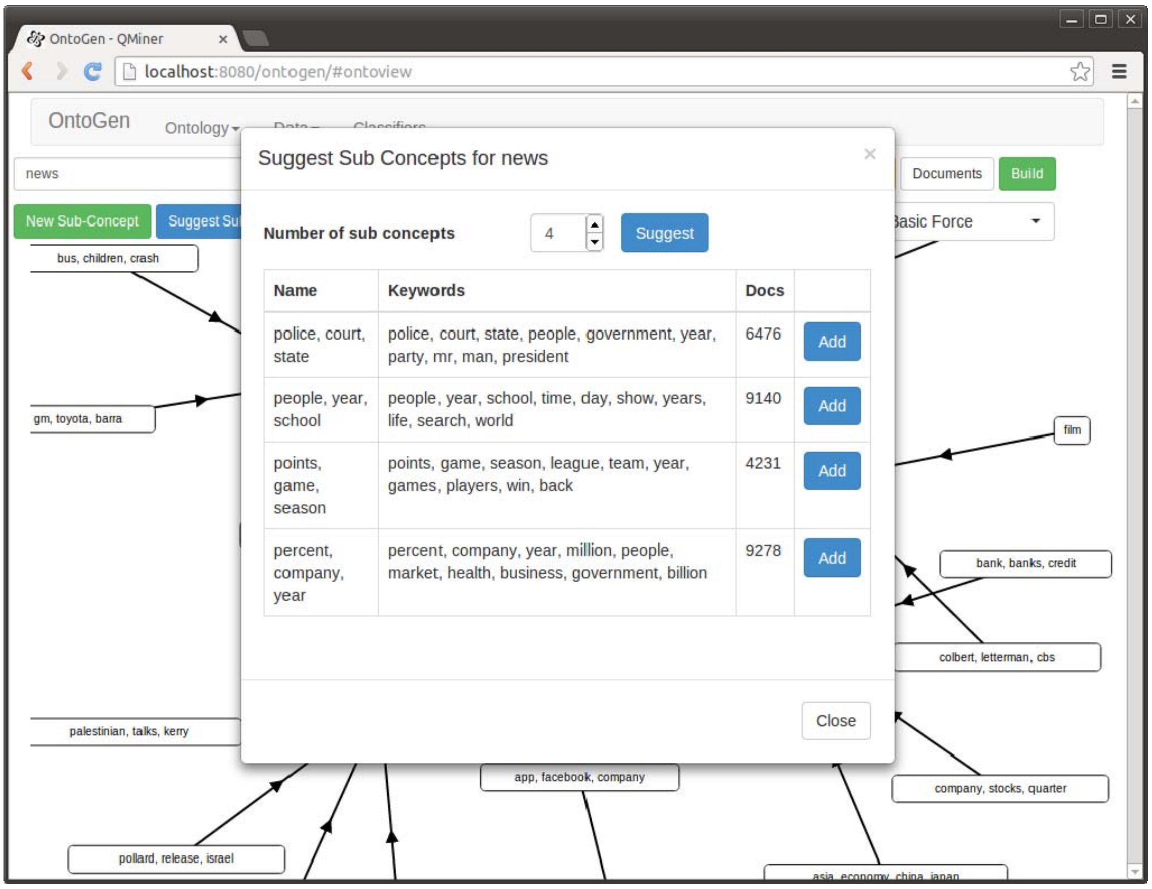
3.1 Unsupervised Methods

We use TF/IDF keyword extraction throughout the application to provide a human readable summary of a set of documents such as those that constitute a concept, which is really just a set of documents grouped together.

Unsupervised concept suggestions are based on the KMeans++ clustering method from [1, 2] where clusters of instances from the selected concept are treated as sub-concept suggestions. The main advantage of the unsupervised method offered is that it requires very little input from the user – only the number of sub-concepts (and even that would be possible to omit in further development of the system is required). Figure 1 shows an example of clustering based sub-concept suggestion in a news dataset.

3.2 Semi-supervised Methods

For semi-supervised method we use SVM based active learning method [4], where the user supervision is provided first by a query describing the concept that the user has in mind and followed by an initial sequence of Yes or No questions. This process is shown in Figure 2. The system refines the suggested concept after each reply from the user and the user can decide when to stop the process based on how satisfied he is with the current concept suggestion (based on keywords and number of documents). A user could start by providing the query *election* on a news article dataset and create a concept that includes all articles related to a specific election, all elections or even politics in general, depending on the answers given to the questions.



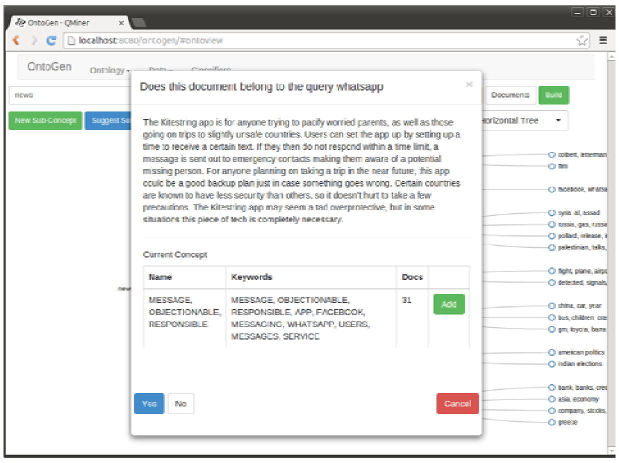
Suggest Sub Concepts for news

Number of sub concepts: 4 Suggest

Name	Keywords	Docs	
police, court, state	police, court, state, people, government, year, party, mr, man, president	6476	Add
people, year, school	people, year, school, time, day, show, years, life, search, world	9140	Add
points, game, season	points, game, season, league, team, year, games, players, win, back	4231	Add
percent, company, year	percent, company, year, million, people, market, health, business, government, billion	9278	Add

Close

Figure 1: C*lustering based sub-concept suggestion.*


Does this document belong to the query whatsapp

The Kitestring app is for anyone trying to pacify worried parents, as well as those going on trips to slightly unsafe countries. Users can set the app up by setting up a time to receive a certain text. If they then do not respond within a time limit, a message is sent out to emergency contacts making them aware of a potential missing person. For anyone planning on taking a trip in the near future, this app could be a good backup plan just in case something goes wrong. Certain countries are known to have less security than others, so it doesn't hurt to take a few precautions. The Kitestring app may seem a tad overprotective, but in some situations this piece of tech is completely necessary.

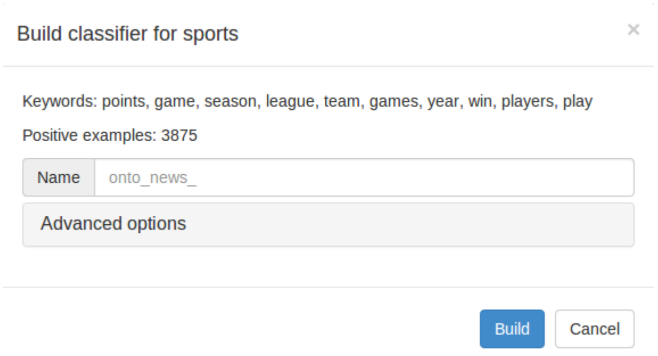
Current Concept

Name	Keywords	Docs	
MESSAGE, OBJECTIONABLE, RESPONSIBLE	MESSAGE, OBJECTIONABLE, RESPONSIBLE, APP, FACEBOOK, MESSAGING, WHATSAPP, USERS, MESSAGES, SERVICE	31	Add

Figure 2: *Active Learning based sub-concept suggestion.*

3.3 Supervised Methods

The user can also select a concept, created manually, semi-automatically or fully automatically, and press a button to build a classifier from it. Figure 3 shows the resulting dialog which allows naming the classifier to be built and allows the user to specify additional parameters, such as classifier hyperparameters, hidden by default.


Build classifier for sports

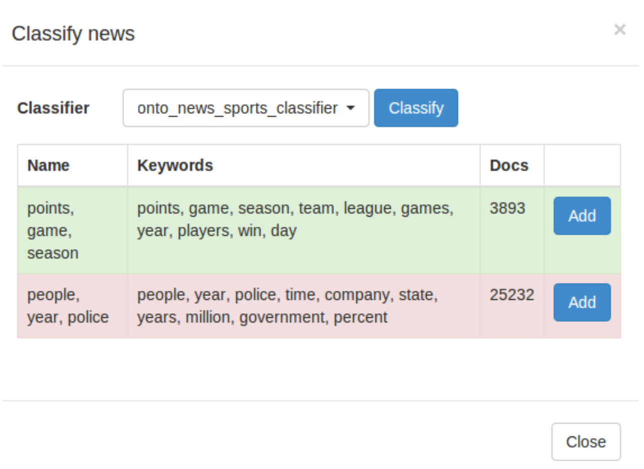
Keywords: points, game, season, league, team, games, year, win, players, play

Positive examples: 3875

Name: onto_news_

Figure 3: *Creating a classifier from a concept.*

This feature allows the user to create concepts in a new ontology based on concepts from a previous ontology or use a classifier built for one concept to create sub-concepts in a different concept. This is shown in Figure 4. Consider a set of classifiers created from an ontology developed on a news article dataset. These could be used to classify datasets of web pages or tweets into the same ontology concepts e.g. *Politics*, *Sports*, etc. Now consider working on a dataset of publicly traded company descriptions. It is possible to have the concept of *Research* as a sub-concept of the concepts of *Energy*, *Medical* and *Technology*. In this case, a classifier that identifies technology companies that are research focused could also be used for indentifying energy or medical companies that are also research oriented.
We use binary classifiers and thus it is possible, though not necessary, to create a positive concept and a negative concept. This divides a set of documents into documents that belong to a concept and concepts that do not. Figure 4 shows the positive concept with a green background and the negative concept with a red background.
Another interesting possibility is to create a concept in the current corpus using the semi-supervised techniques described above, build a classifier for it and use that classifier in a different application thanks to Elycite's ability to provide classifiers as services.


Classify news

Classifier: onto_news_sports_classifier

Name	Keywords	Docs	
points, game, season	points, game, season, team, league, games, year, players, win, day	3893	Add
people, year, police	people, year, police, time, company, state, years, million, government, percent	25232	Add

Figure 4: *Using a classifier to create sub-concepts: positive (green) and negative (red).*

4 EXAMPLE USE CASE

A typical workflow begins with an exploratory phase (1), where the user tries to understand the content, topics and their distribution in the corpus. In the subsequent development phase (2), the user develops models for some or all of the previously discovered topics. The final phase (3) consists of applying, the developed models to new data. Let us now consider the case of analyzing comments from marketing questionnaires, where customers provide feedback in a form of a unstructured text.

1. The analyst identifies topics and subtopics in the comments using clustering, with automatically extracted keywords as summaries for each topic. Using active learning, the analyst can drill down to a particular issue or topic he is interested in. This step is illustrated in Figure 5 where a dataset of 80,000 twitter profiles is explored based on the user's description.

2. The analyst creates classifiers for particular topics and exposes them as services. The process of creating a classifier was shown in Figure 3.

3. Services are integrated with another application to provide real-time monitoring, personalized feedback or promotional offers.

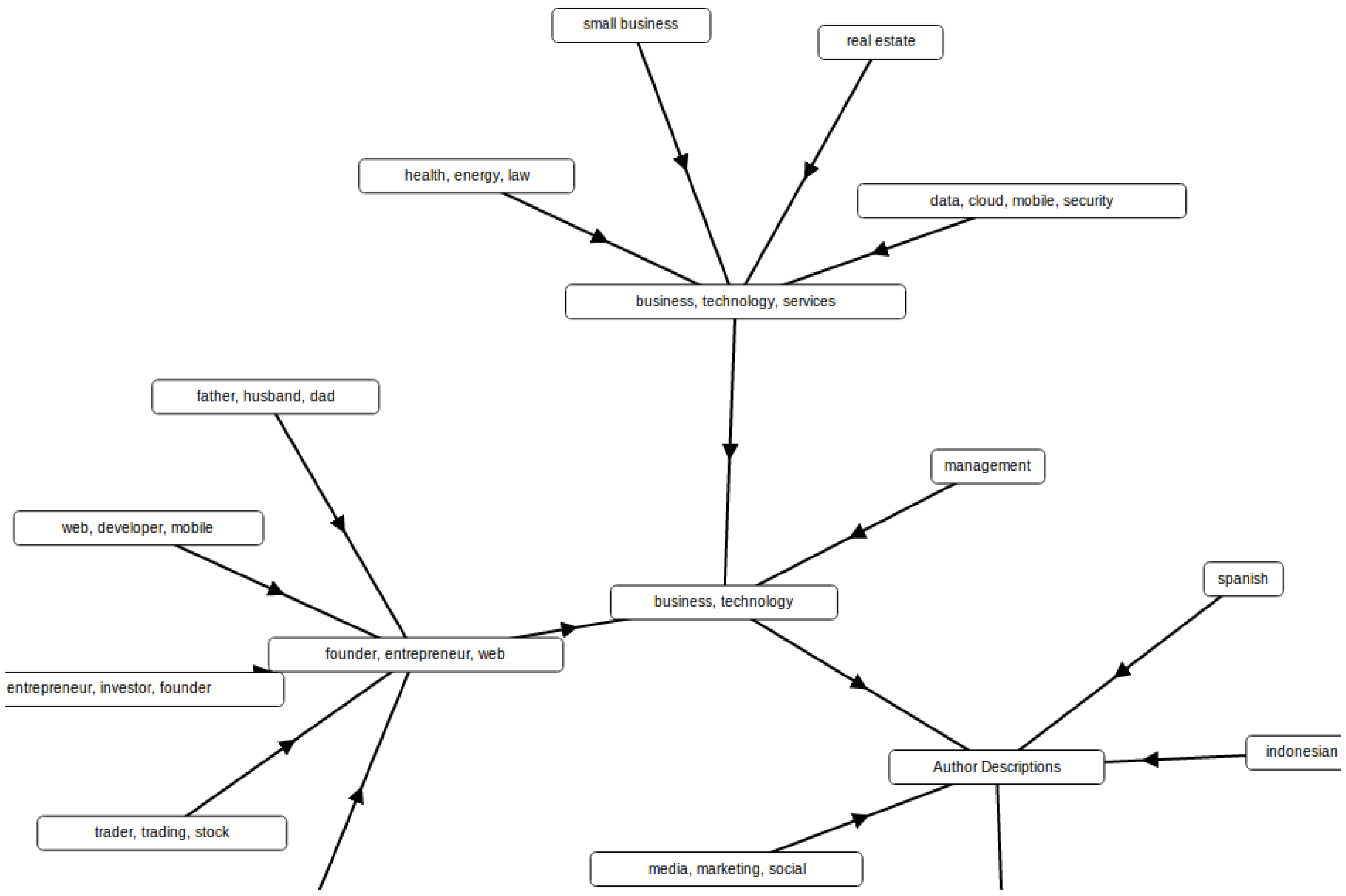


Figure 5: *Visualization of part of an ontology created from a dataset of twitter profile descriptions with Elycite.*

5 FUTURE WORK

Some of the first ideas for future work include addressing the previously mentioned issues related to the interactive use of datasets larger than a few gigabytes and by making incremental improvements in preprocessing such as n-gram generation and the use of the hashing trick [5]. We plan to implement guided learning [6], where the objective is to find documents belonging to a very rare class (i.e. concept) semi-automatically. We also want to provide high quality pre-built classifiers for common tasks such as sentiment classification and topic categorization. The most ambitious part of our planned future work is providing support for numerical, graph, image and time series data. This implies the addition of new preprocessing, feature exatraction or learning, and concept discovery methods and visualizations.

ACKNOWLEDGMENTS

This work was supported by the Slovenian Research Agency and the ICT Programme of the EC under XLime (FP7-ICT-611346).

References

[1] Fortuna, B., Grobelnik, M., Mladenic, D.: OntoGen: Semi-automatic Ontology Editor. Human Interface and the Management of Information. Interacting in Information Environments 4558(Chapter 34), 309–318 (2007)

[2] Fortuna, B., Mladenic, D., Grobelnik, M.: Semi-automatic construction of topic ontologies. Semantics, Web and Mining 4289, 121-131 (2006)

[3] Fielding, R. T., Taylor, R. N.: Principled design of the modern Web architecture. ACM Transactions on Internet Technology 2, 115-150 (2002)

[4] Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. The Journal of Machine Learning Research 2, 45–66 (2002)

[5] K. Weinberger, A. Dasgupta, J. Attenberg, J. Langford, A. Smola: Feature hashing for large scale multitask learning. In Proceedings of the International Conference on Machine Learning (2009)

[6] Attenberg, J., & Provost, F.: Why label when you can search?: alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining* (2010)