Enhancing Business Intelligence with Unstructured Data

MSc Thesis

Extended Summary

Sabrina Masson¹, Helena Galhardas¹ and João Damásio²

¹Institute Superior Técnico, Universidade de Lisboa, Avenida Rovisco Pais, Lisboa, Portugal
²Business Intelligence Department, Link Consulting, Lisboa, Portugal

{sabrina.masson, helena.galhardas}@ist.utl.pt, joao.damasio@linkconsulting.com

Keywords: Structured data, Unstructured data, eXPTO system

Abstract: The growing competition among companies made these realize the importance of knowing their clients as a way to increase their competitive advantage. According what it was referred in (Bhide et al., 2008), 85% of clients data are stored as unstructured data, typically in textual documents, that are not being explored to extract useful information from them. This paper focuses on the problem related with the combination of structured and unstructured data, in a way that both can be simultaneously analyzed and explored to get more information about clients, partners and competitors of a given company. Its resolution led to implementation of the eXPTO system. This system is able to recognize technologies and business clients, partners and competitors of a given company, referred in meeting notes and textual documents written by its employees. The entities that were recognized by this system are converted into structured data. The eXPTO system can be used to explore simultaneously structured and unstructured data, like meeting notes, textual documents or emails. With this system, unstructured data can be analyzed without the necessity of a previous conversion of this type of data into structured data. This system also shows that it is possible to combine inexpensive technologies to solve a typical problem of enterprises.

1 INTRODUCTION

Nowadays, a typical organization stores two types of data: structured data and unstructured data. Over the years, the growing competition among companies made them conscious about the importance of knowing their clients. According to (Bhide et al., 2008), 85% of client data are stored as unstructured data, typically in textual documents, that are not used. Unstructured data are collected by companies, through several interactions with their clients (e.g., email exchanges, meeting notes, etc.). Structured data are also collected by companies. Names, addresses and identifiers of business clients are examples of this type of data. Both structured and unstructured data are usually stored in a CRM (Customer Relationship Management) system. Combining unstructured data with structured data is important because it can lead to business opportunities. When organizations incorporate unstructured data into the decision-making process, they can make better decisions (more timely, more accurate and more informed). Both types of data can be used to infer useful information about the client. Consider, as an example, the CRM system database of a given company. This database contains structured data about companies and the technologies they use. Furthermore, it stores textual data about commercial meeting notes between this company’s employees and its clients. These meeting notes may contain information about competitors and partners that are also present in company’s clients. Data about technologies used by these clients may also be contained in meeting notes. All these types of data are useful, because, for each company, it is possible to know: (i) which competitors of a given company are working with it, (ii) which partners of a given company are working with it and (iii) which technologies are used by it.

Information Extraction (IE) techniques are used to automatically extract structured data from unstructured data (Doan et al., 2012). Two important IE tasks are: named entity recognition and relationship extraction. Named entity recognition (Jurafsky and Martin, 2009) is the task responsible for detecting and classifying all the proper names mentioned in a text. Relationship extraction is the task responsible for finding
and classifying the semantic relationships that exist between the entities recognized in a text (Jurafsky and Martin, 2009). The techniques used in these two tasks are divided in two groups: based on rules and based on machine learning. Regular expressions are an example of a technique based on rules used for recognizing entities (Sarawagi, 2008). Dictionaries are also used for recognizing entities. Dictionaries are defined as a list of terms that we want to find in texts. Machine learning encloses a set of algorithms that can learn from data. The goal of these algorithms is to construct a model based on training data. The generated model is then used to make predictions. Examples of this kind of techniques are: Conditional Random Field (CRF) (Lafferty et al., 2001) and Maximum Entropy Model (McCallum et al., 2000).

Currently, there are technologies that are able to combine structured data and unstructured data, (e.g. InfoSphere Warehouse\textsuperscript{1} and Endeca Information Discovery\textsuperscript{2}). InfoSphere Warehouse converts unstructured data, stored in textual columns of relational databases, into structured data. Endeca Information Discovery reads unstructured and structured data from multiple data sources and load them into a proprietary data model. Full-text search feature is one of the main features of Endeca Information Discovery. Using this feature, the user can look for certain keywords in unstructured data. With Endeca, it is possible to combine structured and unstructured data without having to convert unstructured to structured data previously.

The most relevant research approaches that combine structured with unstructured data are: EROCS (Chakaravarthy et al., 2006), SCORE (Roy et al., 2005) and LIPTUS (Bhide et al., 2007). These systems associate unstructured data stored in files with structured data stored in relational databases. EROCS performs a named entity recognition, by posing a query to a relational database in order to recognize the entities mentioned in textual documents. SCORE performs a SQL query on structured data stored in a relational database and converts the result of this query in keywords. These ones are used to find textual documents related with the query result. LIPTUS extract identifiers from textual documents and use them to find structured data stored in relational databases related with these documents.

The general goal of this paper is to present a proof of concept of a technological solution with similar features to Endeca and InfoSphere, but using Microsoft and open-source technology. Concretely, the first requirement of the solution is to create a software system based on these two technologies to recognize the names of technologies, business clients, partners and competitors of a given company, mentioned in textual notes taken by its employees. The second requirement of the solution is to perform full-text search on unstructured textual data. The entities recognized in textual documents are converted to structured data. The structured data refers to technologies and business clients, partners and competitors of a given company. The unstructured data corresponds to meeting notes, emails and textual documents written by its employees. The third requirement of the solution is to create a system that is able to answer a set of queries. Examples of these queries are: How many are the companies whose meeting notes mention technology SAP BI? Which are the companies whose meeting notes mention NovaBuse? or Which are the companies whose meeting notes mention João Garcia? Some questions like the last one requires the use of keywords in order to perform a full-text search on unstructured data to answer them.

The main contributions of this paper are as follows: (i) the architecture of a software system (eXPTO - eXtraction Per Technologies and Organizations) to address the above mentioned requirements. (ii) the eXPTO entity recognition module that is responsible for recognizing the names of clients, partners and competitors of a given company, as well as technologies, mentioned in textual notes, and stored them in distinct fields of a relational table; (iii) the eXPTO search and data visualization module that is responsible for combining unstructured and structured data; and (iv) a survey of state-of-the-art technologies that combine unstructured with structured data.

We performed an experimental validation of the two eXPTO modules and a comparison between eXPTO and Endeca. From the results obtained, we concluded that the answers we get from both systems, when the set of queries were posed on them, are the same.

This paper is organized as follows. Section 2 presents the most relevant related approaches for combining structured and unstructured data. In Section 3, we describe the eXPTO system that we performed. Section 4 presents the validation of the eXPTO system and we conclude in Section 5 with the main directions for future work.
2 Related Work

This section presents the most relevant approaches that combine structured data with unstructured data. Section 2.1 presents the main commercial enterprise technologies and Section 2.1 presents the research proposals.

2.1 Commercial Solutions

There are several commercial solutions for analyzing and exploring unstructured data in a business context scenario. Commercial enterprise technologies enable, not only the extraction of structured data from text, but also its combination with structured data, stored in relational databases. Two of these enterprise technologies are InfoSphere Warehouse and Endeca Information Discovery. Both technologies support Online Analytical Processing (OLAP) operations and use information extraction techniques to extract structured data from unstructured data.

InfoSphere Warehouse stores unstructured data in textual columns. It uses the UIMA framework to recognize named entities in text. Endeca Information Discovery uses Lexalytics to perform names entity recognition. Lexalytics is a text analytics software that is able to extract entity names, like people, places, companies and products from text. Endeca Information Discovery is also characterized by using a specific data model, called jagged database, to store structured and unstructured data. Endeca Information Discovery supports full-text search, while InfoSphere Warehouse does not. Full-text search is performed over data stored in textual columns of a relational table. This feature enables to explore unstructured data without previously converting it into structured data.

2.2 Research Approaches

There are some research systems that were developed to deal with unstructured data. EROCS (Chakaravarthy et al., 2006), SCORE (Roy et al., 2005) and LIPTUS (Bhide et al., 2007) associate unstructured data, stored in filesystems, to structured data, stored in relational databases.

EROCS, SCORE and LIPTUS associate unstructured data to structured data differently. SCORE is able to dynamically combine textual documents with structured data to get operational business data such as sales or accounting. This system executes a SQL query over the structured data stored in a relational database. The result of this SQL query is converted into a set of keywords. These keywords are provided to a search engine to retrieve textual documents that contain them. SCORE returns, as output, the textual documents retrieved, along with the result of the SQL query. With SCORE a system’s analyst does not need to specify which entities should be recognized in unstructured documents since these are determined by the SQL query posed against the structured data. This is the main advantage of this system.

EROCS was implemented to be a generic solution for named entity recognition. This system receives as input an entity template that represents the entities to be recognized in textual documents. EROCS performs a search on the database to recognize the entities stored therein, which best correspond to noun phrases contained in textual documents. EROCS (Bhide et al., 2008) uses UIMA (Ferrucci and Lally, 2004) to recognize named entities.

Unlike EROCS, LIPTUS was implemented for a specific domain (banking system). LIPTUS intends to be a simple solution able to combine customer interactions (emails and transcribed phone calls) with clients and accounts profiles (client and account data), without compromising the scalability of the system. Structured data correspond to client and account profiles stored in an existing database. The client profile includes attributes such as client name, address and contact. The account profile includes attributes, such as average quarterly balance, date of opening and date of last activity. Textual documents correspond to customer interactions and contain account identifiers and customer identifiers. These identifiers are extracted and then matched with the identifiers present in the customer and account profiles (such as customer ids, credit card or bank account numbers) that are stored in a relational database. The structure of the customer interactions that we want to associate with the account/client profiles is known a priori. LIPTUS also uses UIMA to recognize named entities.

3 The eXPTO System

eXPTO (eXtraction Per Technologies and Organizations) is a software prototype created to answer a set of queries that involves structured and unstructured data, like How many are the companies whose meeting notes mention technology SAP BI and Which are the companies whose meeting notes mention João Garcia?. This system has two distinct and complementary modules: the entity recognition module and the search and data visualization module. The former is responsible for recognizing the names of technologies and business clients, partners and competitors of
the company that are referred in meeting notes and textual documents. The latter is responsible for performing full-text search on meeting notes and textual documents. Furthermore, the search and data visualization module enables to analyze unstructured data without previously converting it into structured data.

The \textit{eXPTO search and visualization module} supports two types of questions: quantitative questions (\textit{how many?}) and qualitative questions (\textit{which??}). Examples of both types of questions are as follows. The following acronyms have as meaning: \textit{TX}: technology’s name; \textit{CX}: company’s name; \textit{CCX}: competitor company’s name; \textit{PCX}: partner company’s name; \textit{CLCX}: client company’s name and \textit{KWX}: Keyword.

**Quantitative Questions (How many??)**
- How many are the companies whose meeting notes mention technology \textit{T1}?
- How many are the companies whose meeting notes mention company \textit{C1}?
- How many technologies were mentioned in the meeting notes of company \textit{C2}?

**Qualitative Questions (Which??)**
- Which are the companies whose meeting notes mention competitor \textit{C1}?
- Which are the companies whose meeting notes mention partner \textit{PC1} and technology \textit{T1}?
- Which are the companies whose meeting notes mention client \textit{CLC1} and keyword \textit{KW1}?
- Which are the companies whose meeting notes mention keyword \textit{KW2} or \textit{KW3}?

### 3.1 General Architecture

Figure 1 represents the architecture of the \textit{eXPTO} prototype. This architecture is composed of: (i) the \textit{CompanyTechnology} relational database; (ii) the entity recognition module; and (iii) the search and data visualization module.

The \textit{CompanyTechnology} relational database contains six tables. The \textit{CompanyType} table stores the three possible company’s types: client, partner and competitor. The \textit{Company} table stores the identifier, name and type of companies to recognize in unstructured data. The \textit{Technology} table stores the identifier and name of technologies to recognize in unstructured data. The \textit{Notes} table is different from the remaining tables and stores both types of data: structured and unstructured. The structured data - stored in the \textit{Notes} table - corresponds to the identifier of the textual document/meeting note, and the name and identifier of the company that owns the textual document/meeting note. The unstructured data corresponds to the textual document/ meeting note of a given company. This table has two textual columns to store them: \textit{TextNote} and \textit{TextNotePhoneme}. The first column stores the textual documents and meeting notes, while the second one stores a text produced by the \textit{entity recognition module} that contains words of textual documents and meeting notes concatenated with the corresponding phoneme (a basic unit of a language’s phonology used to represent the sound of a syllable).

The \textit{Company_Notes} and \textit{Technology_Notes} store the entities recognized in textual documents and meeting notes by the entity recognition module. The former stores the identifier of the company that was recognized in meeting notes and textual documents. The latter stores the identifier of the technology that was recognized in meeting notes and textual documents. Both tables also store the identifier of the textual documents/meeting notes where these entities were recognized.

The \textit{entity recognition module} receives the textual documents and meeting notes, stored in the \textit{TextNote} column of the \textit{Notes} table, as input. It is responsible for recognizing names of technologies and companies. This module also receives, as input, four dictionaries, used to recognize the following entities: Technologies, Clients, Partners and Competitors. These dictionaries store the names of technologies and business clients, partners and competitors of a given company. If the entities recognized by the entity recognition module are companies, this module stores them in the \textit{Company_Notes} table. If the entities recognized are technologies, this module stores them in the \textit{Technology_Notes} table. The \textit{entity recognition module} is also responsible for converting each word of textual documents and meeting notes into phonemes. A new text is created with the word and the corresponding phoneme separated by '@'. Consider, as an example, the following sentence: “\textit{Solicitada apresentação}”. The newly created text will be ”\textit{@Solicitada @ SLTD @ apresentação @ PRSNTS}”. This new text is stored in the \textit{TextNotePhoneme} column of the \textit{Notes} table to be used by the \textit{search and data visualization module}.

The \textit{search and data visualization module} answers questions that involve structured and unstructured data stored in the \textit{CompanyTechnology} relational database. With this module, a user can specify a set of filters and perform a full-text search based on keywords to answer questions about the data stored in the relational database. Filters are used for selecting structured data, while full-text search based on keywords is used for searching unstructured data stored in the \textit{TextNote} column of the \textit{Notes} table. If there is no exact match between the keywords provided by the user and the words occurring in textual documents...
and meeting notes, the search and data visualization module performs an approximate match. The approximate match is performed between the keywords provided by the user and the words and phonemes of the texts stored in the TextNotePhoneme column.

### 3.2 The Entity Recognition Module

Figure 2 presents the architecture of the entity recognition module. This architecture is composed of: (i) the reading module; (ii) the named entity recognition engine; and (iii) the writing module. The UIMA framework was used to implement this module.

The reading module is responsible for establishing a connection to CompanyTechnology database, through a database connectivity driver, to have access to the Notes table and get the meeting note/textual document stored in its TextNote column. The corresponding NoteID is obtained too. NoteID is extracted to identify which meeting notes companies and technologies were recognized. The NoteID and the meeting note/textual document (referred here as TextualNote) is inserted in a data structure which is passed to the named entity recognition engine.

The named entity recognition engine is the engine responsible for named entity recognition. This engine receives, as input, the data structure with the NoteID and the TextualNote that were extracted from the previous module and four dictionaries. These dictionaries are used by this engine to recognize not only business clients, partners and competitors names, but also technologies names that are referred in the TextualNote. This engine uses an algorithm that looks up on the four dictionaries to get the names contained in that files. For each dictionary entry, the algorithm executes an exact or approximate match according to the length of companies or technologies names that forms the dictionary entry. The recognized entities are written in the structured data that receives as input. Now, this data structure includes, not only the NoteID and TextualNote, but also the companies and technologies names recognized by this module. The data structure is passed to the next module which is the writing module.

The writing module is responsible for populating the Company_Notes and Technology_Notes tables, as well as the TextNotePhoneme column of Notes table. If the recognized entity is a business client, partner or competitor of a given company, the writing module inserts, in Company_Notes table, the CompanyID of this recognized company and the NoteID of the TextualNote where this company was recognized. However, if the recognized entity is a technology, the writing module inserts, in Technology_Notes table, the TechnologyID of the recognized technology and the NoteID of the TextualNote where this technology was recognized. The writing module obtains the NoteID from the data structure that receives as input, but the CompanyID and TechnologyID are obtained from the Company and Technology tables accessed by this module. The names of companies and technologies recognized by the named entity recognition engine is passed to the writing module. These names are used by this module to get, from Company and Technology tables, the CompanyID of recognized companies and the TechnologyID of recognized technologies, respectively. The writing module is also responsible for converting each word of each meeting note/textual document from TextNote column into phonemes. The word and the corresponding phoneme are, then, con-
A new text composed of words and the corresponding phoneme, separated by @’s, are created and inserted in TextNotePhoneme column. The phoneme is obtained by applying an algorithm similar to soundex, but for a Portuguese language. This algorithm receives a Portuguese word, as input, and returns, as output, its phoneme. The TextNotePhoneme column is used by the search and data visualization module to perform an approximate match, if there is no exact match between keywords provided by the user and words of meeting notes/textual documents stored in TextNote. The writing module also uses a database connectivity driver to connect to the CompanyTechnology relational database.

### 3.3 Search and Data Visualization Module

The search and data visualization module of eXPTO system was implemented with the Microsoft SQL Server Reporting Services (SSRS) and the full-text search feature provided by Microsoft SQL Server. Figure 3 presents the architecture of the search and data visualization module. The search and data visualization module provides a visualization screen that can be accessed by a user through a web browser. Through this screen, a user is able to specify a set of filters to access to structured data and a full-text search based on keywords to answer queries about data stored in the CompanyTechnology database. The inserted keywords and selected filters are passed to SSRS. This technology was used to create the reports that are visualized in the visualization screen. These reports are the interfaces of the search and data visualization module. SSRS is composed of Report Server and Report Designer. Report Server stores reports developed and deployed by Report Designer. Report Server receives, as input, the selected filters and keywords inserted by a user. Filters specify a set of conditions about the structured data to access. These conditions are converted in a SQL query that is posed on the relational tables of CompanyTechnology database. The keywords that were inserted are used by Report Server to access unstructured data stored in TextNote and TextNotePhoneme columns of Notes table. The TextNotePhoneme column is used in approximate match. This match is applied when the exact match between the keywords provided by the user and the words occurring in meeting notes/textual documents, stored in the TextNote column, are not well succeeded. The full-text search feature of Microsoft SQL Server is used by this module to perform a search over unstructured data stored in TextNote and TextNotePhoneme columns.

### 4 Experimental Validation

This section presents the experiments we performed to validate the eXPTO system. Section 4.1 presents the two types of experiments performed. Section 4.2 presents the experimental setup. Section 4.3 presents the measures we used and Section 4.4 reports the results obtained.

#### 4.1 Experiments Performed

We performed two types of experiments: (i) a validation of each of the two eXPTO modules, and (ii) a comparison of eXPTO and Endeca. The validation of the entity recognition module was performed to check if the companies and technologies defined in the dictionaries were correctly recognized. The validation of the search and data visualization module was performed to check if a given set of questions were correctly answered. We performed two kinds of validation of the search and visualization module: (i) a validation by questions and (ii) a validation based on keywords. The goal of performing a validation by questions was to formulate a set of questions to be executed in the search and data visualization module and compare the obtained results with the expected
The goal of performing a validation based on keywords was to evaluate the eXPTO system’s behaviour when mispelling keywords are inserted by users.

The comparison between eXPTO and Endeca was performed by posing a set of qualitative and quantitative questions (described in Section 4.2) to both systems and then compare the answers obtained. The goal of this validation is to check if the answers returned from both systems are the same.

4.2 Experimental Setup

We used 67 meeting notes and textual documents provided by a given company.

We used 30 questions to validate the search and visualization module. These questions were not posed in natural language. They were translated into filters and/or keywords, using the available interface of the eXPTO system. Examples of these questions are presented as follows:

- How many are the companies whose meeting notes/textual documents mention Link edock-Link?
- How many are the companies whose meeting notes/textual documents mention Casa Pinar?
- Which are the companies whose meeting notes/textual documents mention competitor CGI?
- Which are the companies whose meeting notes/textual documents mention partner HP and technology IBM Lotus?
- Which are the companies whose meeting notes/textual documents mention competitor Oracle and Miguel Vicente?
- Which are the companies whose meeting notes/textual documents mention Portugal or Espanha?

We also inserted misspelled keywords in the search field to validate the approximate match capability of eXPTO.

In the comparison between eXPTO and Endeca, the same set of 30 questions defined for the search and data visualization module was also used. These questions were posed to both systems.

4.3 Validation Measures

The entity recognition module was validated according to the measures: Precision, Recall and F-Measure (Jurafsky and Martin, 2009). Precision is defined as the fraction of retrieved information that is relevant, while Recall is defined as the fraction of relevant information that is retrieved. F1-Measure corresponds to the harmonic mean of both Precision and Recall.

An entity recognized in the text can belong to one of two types: company or technology. Let us define e as a type of entity referred in TextNote column of Notes table. We have:

- True Positives (TPe) as the number of successfully recognized entities of type e;
- False Positives (FPe) as the number of recognized entities that are said to be of type e but are not from type e;
- True Negatives (TNe) as the number of successfully recognized entities that were not of type e;
- False Negatives (FNe) as the number of recognized entities that are of type e but are said to be other type;

Precision, Recall and F1-Measure can accordingly be defined as follows:

\[
P_e = \frac{TP_e}{TP_e + FP_e}, \quad R_e = \frac{TP_e}{TP_e + FN_e}, \quad F_1e = \frac{2P_eR_e}{P_e + R_e}
\]

4.4 Experimental Results

Tables 1 and 2 contain the number of entities classified as true positives, false positives and false negatives, as well as Precision, Recall and F1-Measure for each type of entity: company and technology. The obtained results with the validation of the entity recognition module were quite good. From the

Figure 3: Architecture of the Search and Data Visualization Module.
validation of the entity recognition module, we concluded that companies and technologies identified as false positives result from lexical ambiguity. The ambiguity occurs when the same name corresponds to two distinct entities. This happens with entities like Microsoft and Oracle that, according to the context of the meeting notes/textual documents where it occurs, may be referring to a company or technology. There is also lexical ambiguity between some names of people and companies. The name Bentos, for instance, can match the surname of a person or the name of a given company. Furthermore, we also concluded that companies and technologies identified as false negatives result from the absence of entries in dictionaries and/or suitable names. The company’s name Companhia de Seguros Aoreana, for example, was not recognized by the entity recognition module because the meeting note where this name had occurred refers to Aoreana Seguros and not to Seguros Aoreana (the entry of the dictionary). The results could be better if more meeting notes/textual documents are used to define the entries of dictionaries.

Table 1: Precision, Recall and F1-Measure of the recognized companies in the validation of the entity recognition module.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.895</td>
</tr>
<tr>
<td>Recall</td>
<td>0.971</td>
</tr>
<tr>
<td>F1-Measure</td>
<td>0.932</td>
</tr>
</tbody>
</table>

Table 2: Precision, Recall and F1-Measure of the recognized technologies in the validation of the entity recognition module.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.961</td>
</tr>
<tr>
<td>Recall</td>
<td>0.892</td>
</tr>
<tr>
<td>F1-Measure</td>
<td>0.925</td>
</tr>
</tbody>
</table>

From the results obtained in the comparison between Endeca and eXPTO, we concluded that the answers returned from the set of 30 questions were posed on them, are basically the same. After analyzing these answers, we realized that only two questions returned different answers. These questions were How many are the companies whose meeting notes/textual documents mention Casa Pinar? and How many technologies were mentioned in the meeting notes of SMAS? It happened because Endeca, when using dictionaries, executed an exact match between the dictionaries names and the names mentioned in meeting notes/textual documents. In Endeca, the analysis of unstructured data can be improved, through the use of Lexalytics. However, this engine was not used in this validation.

From the validation of the search and data visualization module, we concluded that keywords inserted in uppercase, lowercase or without accents do not influence the answer that is returned.

5 CONCLUSIONS

There are two types of data: structured and unstructured. Combining these types of data in order to infer useful information about the client is a problem that many companies are facing today. In this paper, we present a survey of state-of-the-art technologies and approaches that are able to combine unstructured with structured data. Furthermore, we present a proof-of-concept of a software system (named eXPTO) with similar functionalities to the commercial state-of-the-art systems using open-source and Microsoft technologies. eXPTO is able to recognize the names of clients, partners and competitors of a given company, as well as technologies, mentioned in meeting notes and textual documents written by its employees. eXPTO is composed of two modules: the entity recognition module and the search and data visualization module. We conducted an experimental validation of eXPTO and the results obtained are promising and confirm it is an alternative to the state-of-the-art commercial systems.

The eXPTO prototype can be further developed in two directions. First, we should resolve the lexical ambiguity problem, whose resolution requires the analysis of the context of meeting notes and textual documents, where companies and technologies are mentioned. Second, we also can improve this system by using more techniques for named entity recognition, in addition to the dictionaries, to recognize the names of companies and technologies.
REFERENCES


