Distributed text mining and natural language processing for DataMiningGrid

Daniel Sonntag
DaimlerChrysler Research and Technology, RIC/AM
89013 Ulm Germany

daniel.sonntag@daimlerchrylser.com

March 9, 2004

Abstract

This paper positions text mining components and text mining challenges in the distributed DataMiningGrid architecture. In distributed text mining, the controlled and intercommutable use of statistical text processing and different level linguistic text processing resources for fast domain adaptation is the main objective. Efficiency of subtasks will be guaranteed by specialised distributed classification methods.

1 Introduction

We define Distributed Computing as cooperation of several computers working together on a particularly processing-intensive problem. A single computer accounts for local processing needs and is linked toward other computers by a communication network. *Grid technology is intended to take the concept of the [Internet/Intranet] one stage further by allowing seamless access and use of distributed computing resources as well as information.*[1] The DataMinigGrid project addresses to develop and deploy data mining applications on the Grid [1]. We draw our attention to the text data mining tools and services for the Grid, especially the procedural and natural language-based resources needed for specific text mining applications such as text classification, document retrieval and question answering implemented on top of the generic Grid datamining facilities .

We first introduce the different distributed data mining tasks as specified in [1], before we declare the distributed text mining tasks on this basis. The most important part follows, to define the set of text mining tools for a specific application on the DataMiningGrid. In the outlook, we will give an overview of even more sophisticated application scenarios for distributed text mining in further Grid projects.

[1] Work in progress by Ewan Klein, Miles Osborne and Lex Holt

2 Distribution of data mining task

In the distributed architecture, algorithm development for tools and services is required for distributed operations and heterogenous data types, such as unstructured natural language text, and images, and structured relational data. For the distribution of data mining tasks three increasingly complex scenarios have been considered, 1.) distributed file access, 2.) distributed classifiers with same classifier model, and 3.) distributed learning. The distributed data mining service infrastructure facilitates dynamic and secure pipelining of data mining operations and provides a highly interactive, intuitive way to define, execute, monitor, and manage a data mining workflow. In the next section, we will adopt this distributed data mining infrastructure for distributed text mining, justify its practice and comment on the distributed text mining scenarios to be considered during the project.

2.1 Distribution of text mining task

Many traditional data mining methods have been developed for knowledge-weak domains and for largely homogeneous data samples. Many of these methods have been successfully used in text mining applications. Implemented as distributed systems, parallels between data and text mining can be even better exploited for text mining purposes: In distributed data mining geographically dispersed systems and heterogenous data are in focus, which meets exactly the requirements of advanced text processing and the nature of the text data in different languages itself. These parallels in data and text mining allow for building grid scale text mining and linguistic services and applications. Distribution of text mining will have the following two forms:

1. Distribution of language and processing resources over geographically dispersed systems.
2. Distribution of computationally expensive text mining processes

Distribution of language and processing resources over geographically dispersed systems. The resource distribution mostly concerns the distributed file access grid scenario. We will adopt the following terminology to refer to special types of NLP components.

- Language Resources (LRs) refer to data-only resources such as lexica, corpora, thesauri or ontologies [2].
- Processing Resources (PRs) refer to resources whose character is principally programmatic or algorithmic, such as text classificators, part-of-speech tagger (POS taggers), named entity recognisers (NERs) or grammatical parsers.

PRs typically include LRs such as a lexicon. For the distributed text mining architecture, both LRs and PRs are to be considered as possible components (confer computational grid vs. data grid). For the distributed text mining architecture, distributed access is required both for LR and PR components. What is often neglected is the fact that PRs and LRs have to be build, updated and maintained. Contrariwise, the knowledge acquisition process, i.e. building domain knowledge such as domain lexica and thesauri, is one of the most important activities in NLP-based (knowledge based) text mining. In a distributed architecture, this resources can be locally build and maintained by local experts, thus adding more flexibility in the knowledge acquisition process for text mining in distributed systems. Local services for text mining resources can be used for text classification by building very special purpose classifiers, e.g. for different languages and relying on the distributed access to a multitude of such classifiers.

Distribution of computationally expensive text mining processes Distributed computation concerns distributed classifiers with same classifier model and distributed learning grid scenarios. We anticipate research in fast, distributed text classification. Distributed classification can be established by building one classifier on a centralised machine and distributing it over a network. As a result, efficient processing of text mining subtasks can be guaranteed by specialised distributed classification methods. We will employ these techniques for more structured text representation, in which the time complexity of classification function is roughly cubic. We also plan to use kernel methods to graph-like data the linguistically preprocessed text data can be casted into .

We try to address both paradigms of distributed systems and evaluate the pros and cons. We envision to implement two special text mining methods for the distributed learning environment and in both systems we plan to use an extension to the usual bags-of-words representation in text classification.

- In the first case, we try to connect to a language-dependent classifier for each supported language on the Grid. A language dependent classifier is meant to use an extension to the usual bags-of-word representation in text classification which will be provided by language-dependent LRs. Hereby *Component Communalities* is an important aspect, which state that families of components such as different language classifiers share certain characteristics [2]. These communalities should be modelled. In our proposed text mining architecture, this modelling plays the central role and is directly related to the PR Management: If a component falls into a certain family of components (the equivalence classes of PRs/LRs), it can be replaced by an equivalent component in the sense of its input/output behaviour or a more specialised component can be selected.

- In the second case, we will use text mining technology that is basically statistical, language-independent and multi-lingual and perform distributed classification and content retrieval based on language and domain independent similarity analysis. In equal measure to the first case, the similarity analysis is meant to use higher level text representations instead of bags-of-word representations. We aim to use the distributed kernel methods developed during the DataMiningGrid project and adapt them to text mining purposes. The adaptation to text data could be realised by the help of so-called *word sequence kernels*, a novel way of computing document similarity based of matching non-consecutive subsequences of words [3]. A research aim will be to test word sequence kernels in language-independent distributed classifiers by using this alternative to the traditional bag-of-words representation providing more structured textual input.

Process pipelines The third contribution to distributed learning could take place on a meta learning level as a challenge in distributed systems. Most current Grid application consist of a single distributed process. Natural language processing, on the other hand, is most of the time serial processing by multiple components. As these processes form a pipeline, the aim is to decide on different NLP pipelines or even learn the process pipeline for individual data instances by mining the master control protocol [4]. The master control of a distributed text mining application decides which processing step is to be performed next and which kind of language resource, if any, is to be applied. Apart from the master control, the communication and synchronisation of PRs and LPs are further topics for the distributed learning environment. We will address these topics at a later project state, that is to say which communication and synchronisation patterns can be exploited and be found automatically.

By distributed text mining, concrete application scenarios will be developed which offer new text processing services (for intra-house processes and customers) by following the distributed processes described in this section. In the remaining of this paper, we will discuss the components we figured out for distributed question answering in DataMiningGrid.

2.2 Applied distributed text mining

The basic text mining functionality in a grid-based data mining architecture is distributed document retrieval. Once this functionality is established, more sophisticated applications can be build. We suggest to enhance document retrieval towards special (text zooming) retrieval tasks as used in question answering systems [2]. In the distributed Grid architecture, both PRs and LRs are loosely coupled together instead of hard-wired and the processing steps are distributed among several computers. This accounts for the complexity of natural language processing that is CPU intensive. The major advantage of distributing the PRs and LRs over a network is that the data collections might be stored and maintained by the responsible business units locally, whereas the core question answering engine resides at the Text Mining lab to be maintained by a linguistic engineer. As a result, potential users can connect to the system via a small network client that does not have to be maintained and does not require expensive high performance computing facilities. The infrastructure for the following two applications will be provided by other DataMiningGrid project partners, e.g. the development of Grid-based data access, retrieval, and manipulation services (extraction, load, join, subset selection, filtering ...) needed to prepare data from heterogenous distributed mining sources.

Distributed document retrieval In many business cases, document retrieval on partner's data collections is desired. These resources cannot be copied, because they change frequently and often access is private for parts of the collection. Consider furthermore that many business text collections are only stored at local business

[2] Find answers to natural language questions by searching large document collections. Unlike information retrieval systems very common today in the form of Internet search machines, question answering systems do not retrieve documents, but instead provide short, relevant answers located in small fragments of text. The answers are located by statistical and natural language processing methods (PRs) which often require language resources (LRs).

units. To get access to the complete collection is as well an organisational problem or an authorisation problem.

By distributing the retrieval engine (and the query), these documents can be retrieved locally and access can also be managed locally by the data provider. Then only access to single data instances, not the complete collection has to be granted. An extension to this procedure for even more restricted access is to retrieve only aggregated data which is generated by the data provider. For example, documents can be fully searched, but when retrieved, critical confidential information is blocked such as bank account numbers. The point is that the user's information need can be satisfied because access was granted by blocking the confidential information not relevant to the query. During the project, we will render the use of distributed document retrieval more precisely.

Question answering Such systems introduce a variety of new text processing components to be integrated into the distributed architecture. Fortunately, quite independent PRs and LRs can be added in a monotonic fashion. Therefore, more sophisticated text mining applications, such as NLP based Question Answering, can naturally and easily be expressed in a distributed Grid architecture. Consider furthermore, that a complex text mining query can often be processed asynchronously and co-ordinating. Modern QA systems define different processing steps for different QA types (e.g. fact-based questions, template-based questions, thematic-oriented questions). Because of the variety of processing steps and components involved and the different possibilities for answering a query, QA can be a complex, composite process, for which the best way of selecting and applying single components is not obvious. At least, data access, data availability and good performances of single QA components cannot be guaranteed for all possible query instances in the normal case [4].

In this application scenario, we suggest to supply a distributed question answering application as advanced document retrieval application. We plan to apply the system to a restricted domain, such as car manuals, that are potentially valuable as a source for question answering systems. One basic question to be answered is if more NLP-intensive (and language dependent) approaches to text mining do fit better into the distributed data mining architecture, or language independent approaches. The suggested question answering application is meant to answer this question in part.

We plan to integrate the following text mining PRs into the DataMiningGrid, *ki* means knowledge intensive.

- General document retrieval engine
- Text zooming, passage retrieval engine (part-of-speech tagger (*ki*), chunk parser(*ki*):
- Terminology extraction tool
- Kernel functions for word sequence kernels

Further on we plan to integrate the following text mining LRs into the DataMiningGrid. These components are knowledge intensive by nature. Most of this components reside at our department, for partner components (*pc*) we could provide the Grid-interface.

- Domain terminology
- Wordnet (online lexical reference system)
- Special domain document corpora
- Several term translation tools (*pc*)
- Tools for ontology learning from text corpora such as *Wortschatz* (*pc*)

3 Summary and outlook

We have discussed text mining components and text mining challenges in the distributed DataMiningGrid architecture and explained how distributed data mining can be turned into distributed text mining. The challenges of component selection for distributed processing and real business applications have been described. We defer the outlook to further challenges that come into account:

- Query decomposition. Many retrieval queries can be decomposed into subqueries. In question answering, for example, in template-based question answering several information needs have to be satisfied in once such as scene descriptions: Who did what when? The corresponding subqueries could be processed in parallel.
- Multimedia mining. We define this as the application of data mining and machine learning algorithms to discover patterns (i.e. knowledge) in corresponding factual data and text data. With the Grid architecture as underlying data mining infrastructure (Grid enabled data mining), the combination of text and data mining to multimedia mining can be exploited. That is learning from data (→ data ontologies) to help process, interpret and learn from text (→ text ontologies) and vice versa. Here, the Grid architecture could be the framework for mining intermedia concepts. The side-effect is to address the research-relevant question, how to efficiently include knowledge (linguistic, ontological, ...) into data mining algorithms.

References

[1] Data Mining Grid Consortium, Data Mining tools and services for Grid Computing Environments (Data Mining Grid), research grant proposal, EU IST FP6 004475, October 2003.

[2] Hamish Cunningham, Kalina Bontcheva, Valentin Tablan, and Yorick Wilks. Software Infrastructure for Language Resources: a Taxonomy of Previous Work and a Requirements Analysis . Technical report, Department of Computer Science and Institute for Language, Speech and Hearing, University of Sheffield, UK, 2000.

[3] Nicola Cancedda, Eric Gaussier, Cyril Goutte, and Jean-Michel Renders. Word-sequence kernels. *Journal of Machine Learning Research*, pages 1059–1082, February 2003.

[4] Daniel Sonntag. Distributed NLP, Java Technologies for Distributed Computing, and ML for Question Answering. to appear, *www.coli.uni-sb.de/~sonntag/DISTRIBUTED_NLP_DRAFT.pdf*, 2004.