Project proposal

PPP – ENS Lyon

M1. 2014

Title	Projet Pensées Profondes (PPP)
Coordinator	Marc CHEVALIER
Members	Raphaël CHARRONDIÈRE, Marc CHEVALIER, Quentin CORMIER, Tom CORNEBIZE, Yassine HAMOUDI, Valentin LORENTZ and Thomas PELLISSIER TANON.
Abstract	*Projet Pensées Profondes* aims to build a natural language question answering framework. It would be done at the ENS Lyon, from September to December 2014, by a team of seven M1 students. This proposal presents our subject and exposes our goals.

1 Introduction

What is the birth date of the first president of the United States? This is the typical question that we will answer quickly and succinctly with an automatic question answering system. This requires four steps.

Understanding the natural language. The input string given in standard English has to be transformed into a normal form.

Querying some database (e.g. Wikidata), using the normal form. Some operations may then be applied, like performing a sort.

Selecting the most relevant results.

Displaying the answer in a convenient fashion.

2 Expected results

The first goal of the PPP is to provide a free, modular and well documented query answering tool that will interest research communities for extensibility features.

We expect to have in December a framework that would allow people to ask questions in plain text through a web user interface, a core module able to transform this question into a normal form and a demo backend module able to answer to historical questions using an existing database.

We expect also to have done some research on natural languages analysis using neuronal networks. These researches may lead, if successful, to a small scientific article.

3 Targeted users

Our main purpose is to allow people to design their own modules and easily include them in the tool. For example, sports modules could be done to answer questions about contests results or upcoming meetings. We intend also to make at least one important module, using Wikidata, in order to provide a demo and attract the Wikimedia community. At a time when intelligent search engine tools are increasingly used, we think it is now or never for the open source community to propose its own product.

We choosed Wikidata for its important community and its huge and well structured database.

On the other hand, we also target the end users of the query answering system. We propose a quick and intuitive way to search information.

4 Concurrent services

Several private companies offer question answering frameworks.

WolframAlpha[1], a tool developed by Wolfram Research, provide a web based service to answer directly questions asked in English. It is well known for its abilities to process mathematical statements. It focuses on historical and scientific facts, such as *When was the French revolution?* or *What is the root of pi?*.

Google Knowledge Graph[2], developed by Google, answer to some search on Google by a related summary. For instance, searching *Leonardo da Vinci* will give a short biography of this person, and display its most famous paintings.

Google Now [3], Siri[4] and Cortana[5], three intelligent personal assistants developed respectively by Google, Apple and Microsoft, provide a mobile based service to answer questions asked in natural language (generally, those of the owner of the mobile device). They focus on practical facts, such as *Where is the nearest restaurant?*.

In a first time, the PPP aims at providing a clever way to browse Wikidata, by giving the user the direct answer to his question, not the link of the related Wikipedia web page. Thus, our software will not be suited to answer such practical questions, although some new modules can be made if there exists a related database. Google Now, Siri and Cortana are therefore not direct concurrent services.

The PPP is closer to the WolframAlpha tool, and to some extent to the Google Knowledge Graph. However, it is a free project, and will be modular and well documented. We hope that it will interest the Wikimedia community, in order to keep it updated in the future. Moreover, we will not focus on mathematical questions, as WolframAlpha does.

5 Software

The goal is to develop a query answering framework able to answer to simple questions with different back-ends.

The software is divided in several components which interact via the HTTP protocol.

The user interface consist of several front-ends. The most natural is a web page, but we can also think about an Android application or a Firefox OS web app. Via this interface, we can submit a query.

The query is transformed into a standard form. This form has to be understandable for a computer and describe the question. This work is done in two ways. There is a classical natural language processing

[1] `https://www.wolframalpha.com/`

[2] `http://www.google.com/insidesearch/features/search/knowledge.html`

[3] `http://www.google.com/landing/now/`

[4] `https://www.apple.com/fr/ios/siri/`

[5] `http://windows.microsoft.com/en-us/windows-8/cortana`

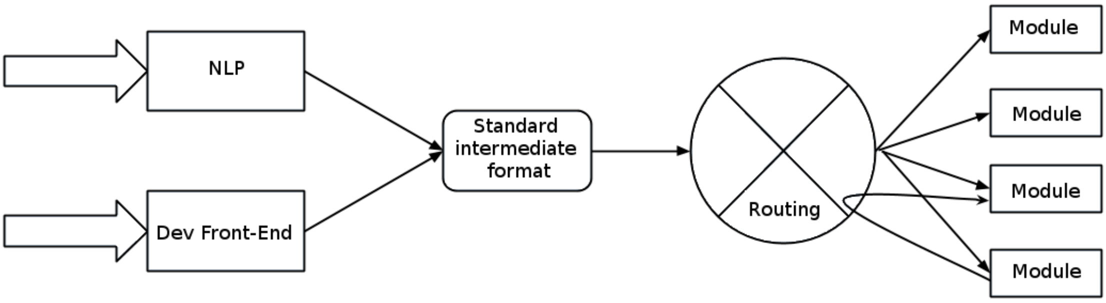

Figure 1: Global structure

module and a machine learning-based module. This kind of transformation is very difficult. To avoid having to go through this step, we can develop another front-end for developers which allow to input a question directly in normal form.

The standard form is sent into the core. The core's job is routing. It has to send the request in each module to obtain the answers. Sometime, a module can request an answer of another module, for example in the question *What is two times the population of Argentina?* which needs a geographic data and a mathematical operation. Such a question needs the successive work of two modules.

To overcome the constraints of language, the core makes some request to the Wikidata module to translate each word of the standard form into the associated Wikidata code.

Once a response is generated, it is sent to the interface to be displayed.

A specific module is needed for each database or kind of computation (mathematical, predictive machine learning. . .).

6 Technological innovations

We will build one of the first open source question answering system. Our goal is to discover the theory of natural language processing, and apply our knowledge to design our tool. We intend to put together some existing libraries (e.g. natural language parsers) and our own implementations of algorithms. In addition, we will implement a neuronal network in order to perform machine learning. For example, we are going to explore deep learning technologies for the natural language processing problem.

One of our most important goal is to provide the first full-modular question answering system. We plan to make available an efficient and attractive system that will motivate people to integrate their own module in the tool. Thus, we think that using the HTTP protocol for the communication between modules is an important part of our project, for the wide variety of programming languages it will allow for the modules.

Finally, our technical approach consists in quickly obtaining first results and a first working tool. Thus, we could use existing libraries in a first time. Then, we will improve it depending on our remaining time.

7 Schedule

We have planned a simple schedule in two periods.

For the mid-term, we plan to have a functional developer front-end, core, the Wikidata module (with the transformation in standard form with Wikidata codes), a first approach of the natural language parser and a base for machine learning. With this base, the software will be able to answer simple questions using the developer interface.

For the final term, we plan to have an enhanced natural language parser (especially for the machine learning-based one), a simple web front-end and a fully operational Wikidata module.

Depending on the remaining time, we may develop other modules, e.g. modules for OEIS, mathematics... Our objective is not to provide a lot of modules, but a modular tool with at least one "demo-module" using Wikidata.

8 Task partition

Several workpackages are defined for this project, with specific goal and participants. They are detailed on the following.

WP1 Global organization (Marc)

Progress and quality of the whole project.

WP2 System administration (Valentin)

Git repositories infrastructure, continuous integration, server management.

WP3 Software architecture (Thomas)

Modular conception of the tool.

WP4 Communication (Tom)

External communication, e.g. website, official announcements.

WP5 Bibliography (Yassine)

Referencement of the used resources, e.g. papers, libraries.

WP6 Router (Valentin)

Other participant: Thomas.

Development of the core module.

Goal for midterm: a working core (end of this workpackage).

WP7 Web user interface (Thomas)

Other participant: Valentin.

Development of the user interface.

Goal for midterm: a basic HTML user frontend, and a developer frontend.

Goal for final evaluation: an improved user frontend, with eventually and Android application.

WP8 Natural language processing (Raphaël)

Other participants: Marc, Quentin, Yassine.

Bibliographical research. Development of the natural language processing module.

Goal for midterm: a first natural language parsing module, using existing libraries.

WP9 Machine learning (Quentin)

Other participants: Marc, Raphaël, Yassine.

Bibliographical research. Development of the machine learning module, to improve the natural language processing.

Goal for final evaluation: usage of the machine learning in language processing.

WP10 Wikidata module (Thomas)

Other participants: Tom, Valentin

Development of the module querying Wikidata.

Goal for midterm: correct answer to questions such as *When was Gandhi born?*

WP11 Add-ons (Marc)

Other participants: Quentin, Raphaël, Thomas, Tom, Valentin, Yassine

Development of other modules, e.g. OEIS, sport.

9 Budget

As demo servers will be provided by Wikimedia, we do not need any particular budget.

10 Test protocol

Evaluating a question answering system is quite a subjective task. Actually, we do not plan to have a full exhaustive test protocol. We will probably evaluate manually the relevance of our tool on some questions. Depending on its performances, we could also use existing benchmarks provided by question answering challenges (e.g. TREC challenge).

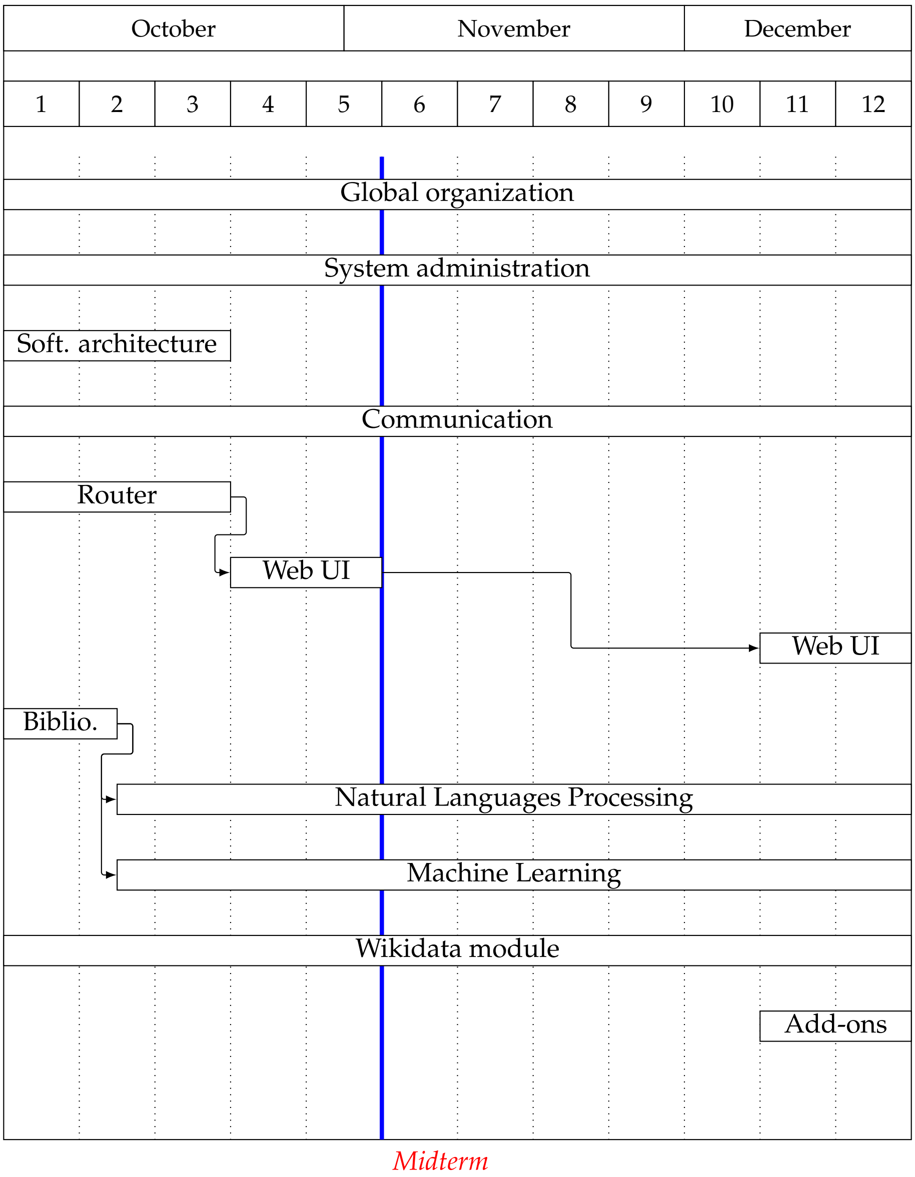

Figure 2: Gantt diagram of the project