POLITECNICO
DI TORINO

Department of Control and Computer Engineering

Master's Degree in Computer Engineering

Master's Thesis

Word Sense Disambiguation technique based on Semantic Networks and Expectation Maximization

Tutor:
Prof. Elio Piccolo

Candidate:
Stefano Zuccato
MAT. 216262

ACADEMIC YEAR 2016/2017

Contents

Abstract

The work presented in this dissertation illustrates the research and development of an application that can provide faster and more reliable word sense disambiguation on all the words of a user input; also, it will only need a narrow context to be provided by the user input, because it will automatically expand its knowledge base. I have done this, while I was working on the topic of Natural Language Processing, for Blue Reply srl.

We will start with an overview of the concept of Word Sense Disambiguation, which is necessary to understand what scientific motivations and thoughts led to the implementation of the project. Subsequently, a brief description of the already-in-use methods will be provided, along with practical examples of their implementations. A system description will follow, with a comprehensive explanation of the employed technologies, and the system's structure and components. Finally, we will provide the performance evaluation, analyzing how the system performed, and how said evaluation was carried through.

Chapter 1

Introduction

This chapter provides an overview of the concepts related to the topic of Word Sense Disambiguation, the motivations and the thought process that led me to the implementation of this idea, as well as the scientific motivations behind this work, and its possible applications.

1.1 Word Sense Disambiguation

Words can have different meanings, and this lexical ambiguity is a fundamental characteristic of any language. Two types of this ambiguity can be found: **polysemy** and **homonymy**. The distinction depends on whether or not the meanings of the considered words are related. The *bark* of a dog versus the *bark* of a tree is an example of homonymy; *review* as a noun and as a verb is an example of polysemy.[1] This does not commonly constitute a problem, because even if in principle words are ambiguous, a human being will find very little ambiguity in this kind of text.[2]

Lexical disambiguation, therefore, is nothing less than determining the meaning of every word in a specific context, which appears to be a largely unconscious process in people,[2] but results to be quite complex for machines, to the point it is actually described as an **AI-Complete** problem[3].

A general definition of this kind of problems is included in the 1991 Jargon File, and states:

> **AI-Complete:** [MIT, Stanford, by analogy with ‘NP-complete’] adj. Used to describe problems or subproblems in AI, to indicate that the solution presupposes a solution to the ‘strong AI problem’ (that is, the synthesis of a human-level intelligence). A problem that is AI-complete is, in other words, just too hard.[4]

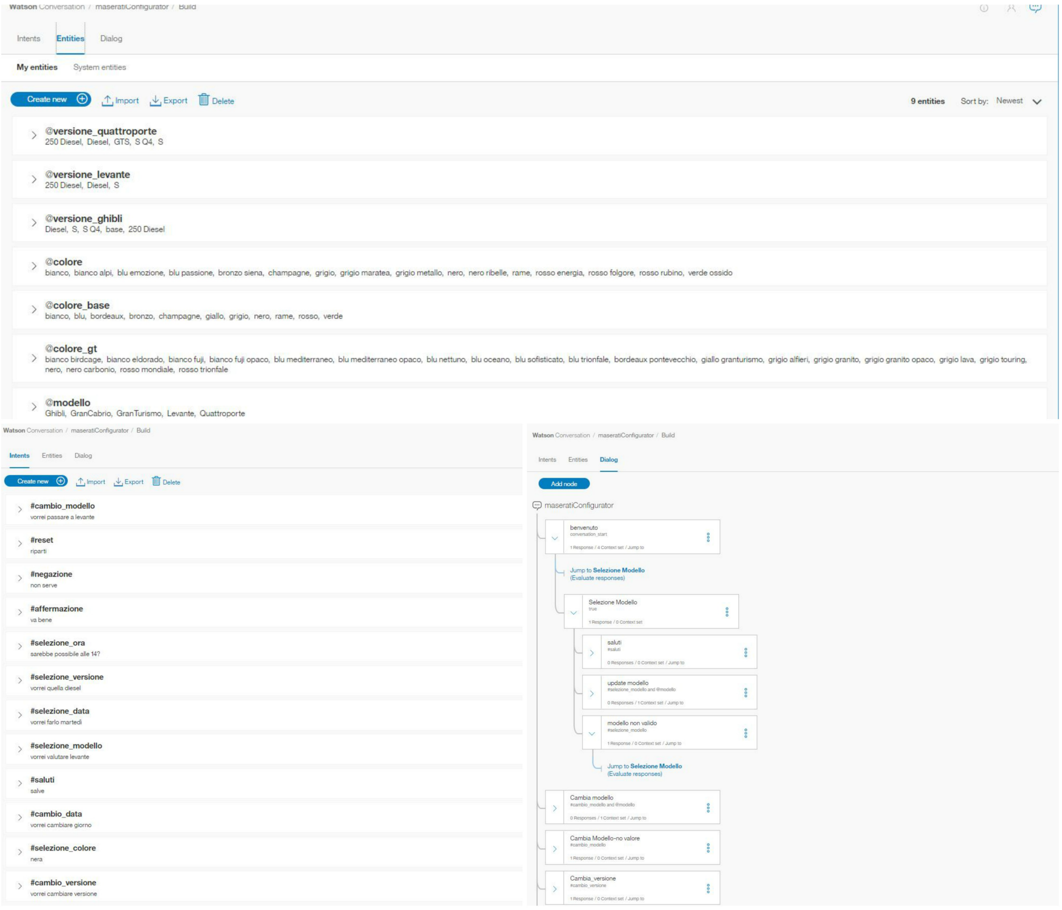
In the field of computational linguistic, this lexical disambiguation is generally referred to as **Word Sense Disambiguation** (WSD), and is defined as the problem of computationally determining which "sense" of a word is activated by using said word in a particular context.[2] The automatic disambiguation of word senses has been an interest and concern since the earliest days of computer treatment of language in the 1950s but, because of its complexity, it still is an open problem.

This task is obviously essential for language understanding applications, such as man-machine communication, and message understanding, but it is sometimes required (or at least helpful) also for applications for which natural language understanding is not a primary aim. For example: *machine translation, information retrieval and hypertext navigation, content and thematic analysis, speech generation and spelling correction.*[2] Some of these applications will be analyzed more in detail further on.

1.2 Introduction to the work and origin of the idea

When I started to work on the thesis at Blue Reply, my focus was on the specific task of creating conversational agents, by using an **IBM** service called **Watson Conversation**. This tool allows developers to create applications capable of understanding a user input in human language, thanks to the employment of machine learning techniques. As a result, the application is then able to simulate a dialog with the user, through a decision graph, navigated using the information retrieved from the input. This service works through a web-based graphical interface1.1, from which the developer is able to create and manage three different objects: *intents, entities and dialog*.

The **dialog** is not fundamental, in order to understand the motivations behind this work, but I will briefly describe it for the sake of completeness: it uses graphical nodes and edges to create a graph that represents the flowchart of the conversation, which the application has to follow, based on the user's input. Each node represents a decision point in the graph: e.g. giving a response to the user, or choosing between two or more branches to follow. This decision is taken by evaluating a series of conditions built on the intents and entities recognized in the text.
How the user input is processed, and what is the information retrieved by this tool, in any case, is the actual key point to understand the origins of this idea. As mentioned before, a developer can define **intents** and **entities**: **intents** are the classes in which an input sentence can be classified, whereas **entities** are concepts (single words or small sentences) that can be recognized in the text by the application.
When an input comes, the service classifies it into one of the defined classes, and returns the corresponding *intent.* Subsequently, it checks the text to see if it recognizes



@versione_quattroporte
250 Diesel, Diesel, GTS, S Q4, S

@versione_levante
250 Diesel, Diesel, S

@versione_ghibli
Diesel, S, S Q4, base, 250 Diesel

@colore
bianco, bianco alpi, blu emozione, blu passione, bronzo siena, champagne, grigio, grigio maratea, grigio metallo, nero, nero ribelle, rame, rosso energia, rosso folgore, rosso rubino, verde ossido

@colore_base
bianco, blu, bordeaux, bronzo, champagne, giallo, grigio, nero, rame, rosso, verde

@colore_gt
bianco birdcage, bianco eldorado, bianco fuji, bianco fuji opaco, blu mediterraneo, blu mediterraneo opaco, blu nettuno, blu oceano, blu sofisticato, blu trionfale, bordeaux pontevecchio, giallo granturismo, grigio alfieri, grigio granito, grigio granito opaco, grigio lava, grigio touring, nero, nero carbonio, rosso mondiale, rosso trionfale

@modello
Ghibli, GranCabrio, GranTurismo, Levante, Quattroporte

#cambio_modello
vorrei passare a levante

#reset
riparti

#negazione
non serve

#affermazione
va bene

#selezione_ora
sarebbe possibile alle 14?

#selezione_versione
vorrei quella diesel

#selezione_data
vorrei farlo martedì

#selezione_modello
vorrei valutare levante

#saluti
salve

#cambio_data
vorrei cambiare giorno

#selezione_colore
nera

#cambio_versione
vorrei cambiare versione

benvenuto
conversation_start
1 Response / 4 Context set / Jump to

Jump to **Selezione Modello**
(Evaluate responses)

Selezione Modello
true
1 Response / 0 Context set

saluti
#saluti
0 Responses / 0 Context set / Jump to

update modello
#selezione_modello and @modello
0 Responses / 1 Context set / Jump to

modello non valido
#selezione_modello
1 Response / 0 Context set / Jump to

Jump to **Selezione Modello**
(Evaluate responses)

Cambia modello
#cambio_modello and @modello
0 Responses / 1 Context set / Jump to

Cambia Modello-no valore
#cambio_modello
1 Response / 0 Context set / Jump to

Cambia_versione
#cambio_versione
1 Response / 0 Context set / Jump to

Figure 1.1: The user interface of IBM's Watson Conversation

any of the defined *entities*, and it does it by trying to find the exact string contained in the entity's definition[1]. *Entities* and *intents* are used by the developer as conditions inside the nodes of the dialog; the application then uses this information to visit the graph, stop at the right node and perform the action contained in it.

My colleagues at Blue Reply and I, though, noticed a limitation in this system: all the entities that the developer wants to be recognized must be explicitly defined, and only the defined entities are recognized. This means that if the user inputs a string that the developer did not consider, but is inherent to the considered domain (which might be useful to the application), said string gets completely ignored. Let's consider this example to make things a little more clear:

example: A developer wants to create a conversational agent capable of providing a set of recipes, based on the ingredients that the user has. To implement this in a

[1] In the latest version of the service (as of May 2017), IBM introduced the possibility of doing a *fuzzy match*, allowing text strings (*entities*) with spell errors, to be recognized as well

service like Watson Conversation, the developer needs to define as entities all the ingredients that he thinks a user can input (meaning that, in order for the user to be able to find them, they have to be present in the application's database). But what if the user wants an ingredient that was not previously considered? Such a system would ignore it completely, because it would not be able to recognize the presence of a possible unknown ingredient. Of course, it might be way more useful to have this information, so to maybe propose an alternative, or simply to acknowledge its absence, to improve the system.

The idea that arose from these considerations, was to create a system capable of analyzing the user's input, and of finding concepts that can, with a certain probability, be associated with a given domain. The work presented in this dissertation is a part of that project: to be more specific, it is its largest and most complicated module. It originates from the need to have a system that can simultaneously disambiguate all the words in a sentence with good precision and at a reasonable speed, and hence recognize a set of high level concepts contained in the input, that will be then classified.

1.3 Scientific motivations

While this project started as a part of a bigger one, it is of interest on its own, for it has a wide variety of applications in almost every NLP application: Word Sense Disambiguation has been drawing more and more attention from recently developed areas, such as bioinformatics and the Semantic Web.[2] This section describes some of its most important applications.

Machine Translation This is the original and most obvious application for WSD. Disambiguation is required for the lexical choice of words that have different translations for different senses, and that are potentially ambiguous within a given domain. An example: a translator needs to translate financial information from English to Italian. The English word *change* could translate to either the Italian word *cambiamento*, which means "transformation", or *monete* which is "pocket money".[2] This sort of translation requires a deep and rich understanding of the source language and the input text, as well as the target language, therefore the problem of automatically producing a good translation of an arbitrary text is far too hard to automate completely.[5] Word Sense Disambiguation is of fundamental aid in understanding which sense of a word is used, and thus finding the proper translation.

Information Retrieval The main goal of Information Retrieval is to help users find information that matches their information needs. Therefore it studies the acquisition, organization, storage, retrieval and distribution of information.[6]
Current IR systems do not use explicit WSD. Instead, they rely on the user typing enough information to contextualize the query and retrieve only the documents relevant to the intended sense. Word Sense Disambiguation can come in help to create more evolute systems, and help to disambiguate queries like *"depression"*, that can look for documents regarding either illness, weather systems or economics.[2]

Information Extraction (IE) and Text Mining This is also known as text data mining or knowledge discovery from textual databases, and refers to the process of extracting interesting patterns or knowledge from unstructured text documents. More generally, the Semantic Web requires automatic annotation of documents, according to a reference ontology[2]. WSD already is of great help for the accurate analysis of the text in many applications, and it is only beginning to be applied in this areas.[2][8].

Even if WSD could bring significant advantages in this wide range of areas, it was not proved to make a decisive difference yet, and therefore is not being widely adopted. This is probably due to the fact that Word Sense Disambiguation is still not accurate enough[2], but the copious amount of research being made, suggests it will soon be a crucial instrument for Computer Science.

[2] "A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are presumed to exist in some areas of interest, and the relationships that hold them (Genesereth & Nilsson, 1987). A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge-based system, or knowledge-level agent, is committed to some conceptualization, explicitly or implicitly. An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an ontology is a systematic account of Existence."[7]

Chapter 2

State of the art

This chapter will describe the main Word Sense Disambiguation approaches, together with the most used techniques for each one of them, giving some practical examples of implementation. The concepts introduced in the first part of the chapter will then be used to contextualize the work presented in this dissertation.

2.1 Main Word Sense Disambiguation techniques

Usually, Word Sense Disambiguation approaches are classified based on the source of knowledge used in sense differentiation. There can be two variants of a WSD task:

- **Lexical Sample Task**: a small pre-selected set of target words needs to be disambiguated in a text;
- **All-words task**: every word in an entire text needs to be considered.

Different techniques are more suited for either the *Lexical Sample*, or the *All-words* task. For example, an approach with supervised machine learning is more suitable for the first type of task, because a classifier can be trained for each word. In the second case, instead, because of the intrinsic data sparseness of the problem, this cannot be done.

Classifying the main WSD approaches is not always easy, but usually they fall into one of the following categories:

- **Knowledge-based** approaches: methods relying on a knowledge-base, like a Thesaurus or Dictionary;
- **Unsupervised Machine Learning** methods: generally *Corpus-Based*; a corpus is a collection of texts that are selected and organized to facilitate linguistic analysis;

- **Supervised Machine Learning** methods: this kind of approaches is *Corpus-Based*, as well as the ones above;
- **Combinations** of the aforementioned methods: all the techniques that use some kind of hybrid approach, exploiting characteristic features of two or more of the above approaches.[2]

While these categories may require a dedicated dissertation, we will now discuss their main characteristics: a basic understanding of the subject is necessary for a thorough comprehension of the work we developed.

2.1.1 Knowledge-based approaches

These methods rely primarily on external resources, such as dictionaries, thesauri, and lexical knowledge-bases. One of the most important techniques in this class of approaches was developed by Michael Lesk in 1986:

The Lesk Algorithm This approach identifies senses of words in context, by using definition overlap. This relies on the fact that a given word in a text and the other words surrounding it, usually share a common topic.
This algorithm works in three steps:

1. Retrieve from a machine-readable dictionary the definition of the senses of the words that need to be disambiguated;
2. Compute a score by determining the overlaps in the definitions for all the possible combinations of senses;
3. Choose the senses that lead to the highest score, i.e. the senses that lead to the highest overlap.

A renowned example exploited by Michael Lesk himself in its original paper, will properly clarify the subject:

The context we are trying to disambiguate is "*pine cone*". What follows, is a list of the possible definitions for each of the two words:

- PINE
    1. kinds of **evergreen tree** with needle-*shaped* leaves
    2. waste away through sorrow or illness

- CONE
    1. solid body which narrows to a point
    2. something of this **shape** whether solid or hollow
    3. fruit of certain **evergreen trees**

The next step will be to compute the overlaps (i.e. the number of words in common) between the definitions:

$$Pine\#1 \bigcap Cone\#1 = 0$$
$$Pine\#2 \bigcap Cone\#1 = 0$$
$$Pine\#1 \bigcap Cone\#2 = 1$$
$$Pine\#2 \bigcap Cone\#2 = 0$$
$$Pine\#1 \bigcap Cone\#3 = 2$$
$$Pine\#2 \bigcap Cone\#4 = 0$$

The above computation shows that the greatest score (2) is achieved by the combination of sense 1 of *Pine* and sense 3 of *Cone*.

This approach appears to be perfectly working, yet problems arise when the task requires to consider more than two words in a context. The next example will illustrate what happens in this case:

Consider the following sentence:

I saw a man who is 98 years old and can still walk and tell jokes

In this context, there are nine words with ambiguous meaning. Here are said words, along with the number of possible senses - included in brackets:

$$see(26), man(11), year(4), old(8), can(5), still(4), walk(10), tell(8), joke(3)$$

The application of the regular Lesk algorithm would lead to an amount of possible combinations equal to:

$$26 * 11 * 4 * 8 * 5 * 4 * 10 * 8 * 3 = 43,929,600$$

The above example shows how the computational complexity of the Lesk algorithm explodes as the number of words in a context increases.
To solve this problem in lexical sample tasks, hence in problems where there is a small set of words (or even one) to be disambiguated, a new version of the Lesk algorithm, called **Simplified Lesk**, was introduced by *Kilgarriff & Rosenweig* in the year 2000. This new version of the algorithm works in the following manner:

1. Retrieve from a machine-readable dictionary all sense definitions of the word to be disambiguated;
2. Determine the overlap between each sense definition and the current context;
3. Choose the sense that leads to the highest overlap.

What follows is an example of application of the *Simplified Lesk* algorithm:

The task is to disambiguate the word *PINE* in the sentence:

Pine cones hanging in a tree

The possible senses for the word *PINE* are:

1. kinds of evergreen **tree** with needle-shaped leaves
2. waste away through sorrow or illness

Applying the Simplified Lesk algorithm we have:

$$Pine\#1 \bigcap Sentence = 1$$
$$Pine\#2 \bigcap Sentence = 0$$

From the above computation we can see that the sense with the maximum overlap is sense number 1.

The initial evaluation of the Lesk algorithm performed by M. Lesk on short samples of text of a manually annotated set, with respect to the Oxford Advanced Learner's Dictionary, was between 50% and 70% of correctly predicted senses.
The Simplified version of the Lesk algorithm turned out to be not only faster and computationally lighter, but also more accurate. In fact, an evaluation performed in 2004 on the data from Senseval-2, with back-off to the most frequent sense, provided proof that the above mentioned performing difference (in regards of accuracy) exists, as the following data highlight:

- Original Lesk: 42%
- Simplified Lesk: 58%

[9][10]

2.1.2 Supervised Machine Learning approaches

The supervised approach to Word Sense Disambiguation uses semantically annotated corpora - that are texts in which words are tagged in context with their sense - to train machine learning algorithms. From these corpora, a feature vector[1] is extracted for every occurrence of the target word, each feature capturing some useful property of the context of the word.
All Supervised Machine Learning methods need in input:

- a word w in a text window d (which we will call a "document")
- a fixed set of classes $C = \{c_1, c_2, ..., c_j\}$
- a training set of m hand-labeled text windows, again called "documents" $(d_1, c_1), ..., (d_m, c_m)$

As a result, they provide a trained classifier $\gamma : d \rightarrow c.$, as an output.
Any kind of classifier can be used. Her are some common examples:

- **Naive Bayes**
- **Logistic regression**
- **Neural Networks**
- **Support-vector machines**
- **k-Nearest Neighbors**

What follows, is an example of an application of Naive Bayes to Word Sense Disambiguation:

We want to disambiguate the meaning of the word "bass". This word has two senses, so it could refer to a "guitar" or to a "fish", hence the resulting classifier can classify the input text in two classes, that we will call:

- f to identify the class associated with the sense of "bass" as "fish"
- g for the class associated with "bass" as "guitar"

We will then use, as training documents and as a test document, sets of neighbor words of "bass" in some sentences. We will have:

[1] a *feature vector* is a simple representation of an object, in this case an instance of a target word, containing feature/value pairs representing some useful information, e.g., the window of words around the target

	Doc	Words	Class
Training	1	fish smoked fish	f
	2	fish line	f
	3	fish haul smoked	f
	4	guitar jazz line	g
Test	5	line guitar jazz jazz	?

The information needed to apply Naive Bayes, which can be retrieved from a tagged corpus like SemCor, is:

- $\hat{P}(c)$ is the **prior probability** of the sense (class) c. We can compute it with the formula:

  $$\hat{P}(c) = \frac{N_c}{N}$$

  where N_c is the number of documents in the training set that are classified with the sense c, and N is the total number of documents in the training set.

- $\hat{P}(w|c)$ is the **conditional probability** of a word in the vocabulary, given a particular sense:

  $$\hat{P}(w|c) = \frac{count(w, c) + 1}{count(c) + |V|}$$

  where $count(w, c)$ is the number of times the word w appears in documents tagged with the sense c, $count(c)$ is the number of words appearing in all documents with the sense c, and $|V|$ is the cardinality of the considered vocabulary.

In this example, the prior probability of the two classes f and g is as follows:

$$P(f) = \frac{3}{4}$$
$$P(g) = \frac{1}{4}$$

And, considering as our vocabulary the set:

$$V = fish, smoked, line, haul, guitar, jazz$$

we can compute the conditional probabilities as follows (I will only report those of the words in the test document):

$$\begin{aligned}
P(line|f) &= \frac{1+1}{8+6} = \frac{2}{14} \\
P(guitar|f) &= \frac{0+1}{8+6} = \frac{1}{14} \\
P(jazz|f) &= \frac{0+1}{8+6} = \frac{1}{14} \\
P(line|g) &= \frac{1+1}{3+6} = \frac{2}{9} \\
P(guitar|g) &= \frac{1+1}{3+6} = \frac{2}{9} \\
P(jazz|g) &= \frac{1+1}{3+6} = \frac{2}{9}
\end{aligned}$$

Now, we can compute the probability of each class given the test document $P(c|d5)$, and then choose the best one:

$$\begin{aligned}
P(f|d5) &\propto P(f) * P(line|f) * P(guitar|f) * P(jazz|f)^2 \\
&= \frac{3}{4} * \frac{2}{14} * \frac{1}{14} * \frac{1}{14}^2 \approx 0.00003 \\
P(g|d5) &\propto P(g) * P(line|g) * P(guitar|g) * P(jazz|g)^2 \\
&= \frac{1}{4} * \frac{2}{9} * \frac{2}{9} * \frac{2}{9}^2 \approx 0.0006
\end{aligned}$$

Now that the model is created for the classifier, the same procedure applied to the test document can be applied to every document that needs to be disambiguated.[5][11]

2.1.3 Unsupervised Machine Learning Approaches

This category of approaches disambiguates word senses without any kind of supporting tool, such as dictionary and thesauri, and also without a labeled training text.

Not using lexical and knowledge resources can usually result in a loss of accuracy, but it can provide really important and useful advantages in terms of versatility and portability. For example, a method relying on such resources cannot easily be ported to other languages, because they need to be replicated for all possible languages, resulting in a great employment of computational resources. Furthermore, the use of fixed sense repositories is constrained to the fixed number of senses present in that repository, which prevents the possibility of discovering new senses of words.

The term *unsupervised WSD* is a little ambiguous itself. Generally, every approach that does not use any sense-tagged corpora is termed as unsupervised, but this definition includes approaches which may use manually-created lexical resources, other than sense-tagged corpora, such as Wordnet, multilingual dictionary, etc. On the other hand, another definition can be considered, which includes only approaches that do not use any of the above mentioned resources. In this case, the word senses that need to be disambiguated in the task are not labeled, which means that we cannot, for example, label the senses of the word *"chair"*, and say *"chair/furniture"* or *"chair/person"*. The only thing that can be done is to cluster (or group) the contexts of an ambiguous word into a number of groups, and then *discriminate* between these groups, without actually labeling them. A wide variety of approaches have been proposed over the years. Here are some of the most relevant:

- **HyperLesk**: a graph-based approach proposed by *Veronis* in 2004. This approach was developed primarily for Information Retrieval applications, and it was meant to identify the paragraphs with the relevant sense of the target word;

- **PageRank**: another graph-based approach, proposed by *Mihalcea et al.* in 2004. It uses Wordnet as a sense inventory. It also uses a Wordnet based semantic similarity measure, which makes it knowledge based. The fact that it does not use any sense-tagged corpora for building a model, makes its categorization as an unsupervised WSD method reasonable;

- **Graph connectivity measures for unsupervised WSD**: proposed by *Navigli* and *Lapata* in 2007. This method requires a graph to be constructed for every sentence using Wordnet; graph connectivity measures are then used to assign senses to the words in the sentence. For each sentence, a set of all possible senses of all words is determined using the sense inventory. Each sense becomes the node in the graph for that sentence. The set of nodes represents all possible meanings the sentence can have. At this point, an algorithm similar to a Depth first Search is initiated;

- **Disambiguation by Translation**. All the above approaches use the untagged corpus of only one language with some knowledge resources to perform the disambiguation. This set of approaches, instead, uses untagged word-aligned parallel corpora in two languages. This is a really interesting approach, because translations are a very strong clue for disambiguation: looking at the translation of a given polysemous word, one is able to restrict the number of its possible senses to the intersection of senses of the target word and its translation.[12]

2.2 Contextualization of this work in the WSD field

Given the previous description of the approaches adopted in the field of Word Sense Disambiguation, and of some of the most important techniques implemented for each one of the described approaches, I can now properly contextualize the work presented in this dissertation, and give a more accurate overview of the project, before diving into a detailed description of it in the next chapter:

The work is an **all-word** WSD task, meaning that the system is trying to disambiguate entire sentences, and not only single specific words.

This project can be classified as a Knowledge-based approach, but, because the content of the input sentences is not known in advance, it is not possible to create a knowledge-base to work with, in advance. The knowledge-base is automatically and dynamically created by the system during its execution, using on-line resources.
During the initial brainstorming for this project, I was keen to use the Lesk algorithm, because of its simplicity and accuracy, but have immediately realized that this system was developed for all-word tasks, which means that the original implementation of the algorithm would require a really long computational time (as previously discussed). This is a big constraint, especially in the context for which this work was developed, that is the interaction with users in human-like conversations. In this field of application, the replies to a user interaction need to happen in a reasonable amount of time.
The use of the simplified Lesk was not possible either, because most of the time, the input from the user is too short, or it comes with a lack of context, making it really difficult to apply the simplified Lesk with good results. In order to solve the problem, I created a new technique, that significantly increases the amount of context to work with (as explained in the next chapter). Also, I used the original implementation of the Lesk algorithm as a score function between two senses of two words; said score function was then used in a statistical algorithm called Expectation Maximization.

Chapter 3

System Description

This chapter firstly provides an introduction and description of the technologies employed in this project; it then explains in detail the structure of the system and of all of its components, describing the flow of information that take place in the program.

3.1 Adopted Technologies

Some important technologies were employed in the development of this project, for different purposes. The language of choice, selected for its peculiar characteristics, is *Scala* with its building tool *SBT*. As a semantic network, I decided to use *BabelNet*, that maps *WordNet* entries with *Wikipedia*, allowing the system to potentially work with a wide variety of languages. To perform POS tagging in the text (further explanations on the subject will be provided later on), I adopted a set of human language tools called *CoreNLP*, developed at Stanford University, together with its Italian version *Tint*. Finally, the chosen algorithm to perform the actual Word Sense Disambiguation on the input text is a custom version of a statistic algorithm called *Expectation Maximization*.
I will now analyze these technologies in detail.

3.1.1 Scala and SBT

Scala fuses object-oriented and functional programming into a statically typed programming language. It was designed to work well with Java and C#, and to adopt a large part of the syntax and type systems of these languages. Scala is not a superset of Java, though: some features are missing, and others are re-interpreted to provide better uniformity of concepts.

Scala programs resemble Java programs in many ways and they can seamlessly interact with code written in Java. Here is an example of code written in both languages:

Listing 3.1: a simple program in Java and Scala

```
// Java
class PrintOptions {
  public static void main(String[] args) {
    System.out.println("Options selected:")
    for (int i = 0; i < args.length; i++)
      if (args[i].startsWith("-"))
        System.out.println("  "+args[i].substring(1))
  }
}

// Scala
object PrintOptions {
  def main(args: Array[String]): unit = {
    System.out.println("Options selected:")
    for (val arg <- args)
      if (arg.startsWith("-"))
        System.out.println("  "+arg.substring(1))
  }
}
```

This snippet of code shows how both languages use the same primitive class **String**, call the same methods, and also use the same operators and conditional control construct. But it also shows some differences between the two languages:

- Scala has an object definition besides the class definition. An object (defined with the keyword **object**) defines a *singleton object*, that is a class with a single instance;

- Java uses the *type id* format for definitions; Scala uses *id : type*, instead;

- Every definition in Scala starts with a keyword (**def main** in the example);

- There is no need to put a semicolon at the end of statements, in Scala;

- Arrays in Scala inherit from Functions, therefore there is no need for special syntax in their definition or access;

- There is no distinction between statements and expressions in Scala: every Function returns a value, and if some code is written in a block, the evaluation of the last expression in it is returned as value. This might simply be the value **{}** which is called **unit**, in Scala;

- Scala has no traditional for-statement; in its place, it has for-comprehension (which Java has for Lists), that allows a developer to iterate directly over the elements of an array, without indexing. The Scala version of for-comprehension is less restrictive than Java's.

Every value is an object in Scala, and every operation is a method call. This characteristic is called *uniform object model.* Every class in Scala inherits from **Scala.Any** and falls into two categories: *value classes* inheriting from **Scala.AnyVal**, and *reference classes* inheriting from **Scala.AnyRef** - that in a Java environment is identified as the class **java.lang.Object**. Every operation is executed with the invocation of a method; for example, the addition **x + y** is interpreted as **x.+(y)**: this Scala feature allows a user to define its own operation, and even to redefine the standard ones. Operations on class members are also treated as method calls in Scala. For instance, when one defines a variable **var x: T** in a class, Scala automatically defines setter and getter methods that are then called every time the variable is referenced, or a new value to it is assigned. The two created methods look like this:

```
def x: T
def x_= (newval: T): unit
```

This way of treating variable access in Scala, makes it possible to define *properties* (in the C# sense).

Scala is also a functional language, in the sense that functions are first-class values. To illustrate the use of functions as values, let's consider the following function, which tests whether a given array has an element satisfying a given predicate:

```
def exists[T](xs: Array[T], p: T => boolean) = {
  var i: int = 0;
  while (i < xs.length && !p(xs(i))) i = i + 1;
  i < xs.length\\
}
```

Both the array's type and the predicate's type are arbitrary. Such arbitrariness is expressed by the type parameter [T] (array's type), and by the *function type* T => boolean, which considers all functions from T to boolean (predicate's type). Function parameters can be applied as normal functions.

Functions in Scala can be nested, as shown in the following function *forall*, that applies a double negation on the previously defined one:

```
def forall[T](xs: Array[T], p: T => boolean) = {
  def not_p(x: T) = !p(x);
  !exists(xs, not_p)
}
```

Functions can also be defined without a name. This allows for slanderer implementations of code, as shown in this shorter version of *forall*:

```
def forall[T](xs: Array[T], p: T => boolean) =
  !exists(xs, x: T => !p(x));
```

Here, *x: T => !p(x)* defines an anonymous function that maps its parameter x of type T to !p(x).
An example of use of the previously defined functions, is in the following snippet of code:

```
def hasZeroRow(matrix: Array[Array[int]]) =
  exists(matrix, row: Array[int] => forall(row, 0 ==));
```

In the expression *forall(row, 0 ==)* the == method of the number 0 is passed as an argument corresponding to the predicate parameter p.
If methods are values and values are objects, then methods are objects. In fact, in general, the *n*-ary function type *(T1, T2, ..., Tn) => T* is interpreted as the parametrized class type Scala.Function*n*[*T1, T2, ..., Tn, T*].

An important issue to deal with, while working in component systems, is how to abstract from required components. There are two principal forms of abstraction in programming languages:

- parametrization, that is the typical form of functional languages;
- abstract members, that is typically adopted by object-oriented languages.

Scala supports both styles of abstraction uniformly, for types as well as for values. Types and values can be parameters, and both can be abstract members.

Scala allows decomposition of objects with pattern matching. Usually, in an object-oriented language, the implementation of structured data would be made by a set of classes representing the various structural constructs. For inspecting structured data, a programmer can solely rely on the methods provided by such classes.
Scala, on the other hand, provides a natural way for dealing with this kind of issues,

by providing the programmer with a mechanism for creating structured data representations similar to algebraic datatypes, and a decomposition mechanism based on pattern matching.

For this purpose, classes with the **case** modifier automatically define a constructor with the same arguments as the primary constructor. On top of this, Scala provides pattern-matching expressions, in which is possible to use such constructors of case classes and patterns. The following code, for example, shows a possible Scala implementation of a λ-term used in λ-calculus, that uses case classes:

```
trait Term;
case class Var(name: String) extends Term;
case class Fun(arg: String, body: Term) extends Term;
case class App(fun: Term, arg: Term) extends Term;
```

With this definition, it is possible to create a λ-term simply by calling the constructor associated with the case classes:

```
Fun("f", Fun("x", App(Var("f"), Var("x")))).
```

At this point, a λ-calculus interpreter can be easily implemented using Scala's pattern-matching:

```
object Interpreter {
  def eval(term: Term): Term = term match {
    case App(fun, arg) => eval(fun) match {
      case Fun(par, body) =>
        eval(subst(body, par, eval(arg)));
      }
      case _ => term
    }
  private def subst(t: Term, v: String, t0: Term) =
    t match {
      case Var(str) =>
        if (str == v) t0 else t
      case Fun(str, t1) =>
        if (str == v) t Fun(str, subst(t1, v, t0))
      case App(f, t1) =>
        App(subst(f, v, t0), subst(t1, v, t0))
    }
}
```

[13]

SBT is an interactive building tool developed for Scala, that can also be used with Java and other languages. It requires very little configuration and, having a Scala-based build definition, it can exploit the full flexibility of the Scala code. It has the ability of doing continuous compilation and testing, and allows parallel task execution. It supports mixed Scala/Java projects, it packages and publishes *jars*, and it generates documentation with *scaladocs*.[14]

Conclusions. This brief description of some of the main features of Scala, now allows me to explain the reason why I chose this language. Considering that the system of which this work is part can potentially work with a huge amount of data in a distributed environment, the functional nature of Scala allows this project to be easily scaled and ported to work in parallel, without losing the ability to interact with some really powerful Java APIs and Libraries for Natural Language Processing. The functional nature of Scala also allows some optimized functional-like data transformations, that are very useful when working with huge data structures.
Finally, the pattern matching capabilities of Scala, that allow for an easy implementation of some paradigms like the λ-calculus (which are very useful in NLP), provided a further reason to choose this language.

3.1.2 BabelNet

BabelNet is both a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms, and a semantic network which connects concepts and named entities in a very large network of semantic relations, made up of about 14 million entries, called Babel synsets. Each synset represents a given meaning, and contains all the synonyms expressing that meaning in a range of different languages.
BabelNet is provided with a Java API, that can be used as a Java Binding to an online HTTP RESTful service, or as a standalone resource downloading the BabelNet indices. BabelNet is also provided as a SPARQL endpoint and a Linked Data Interface, as part of the Linguistic Linked Open Data Cloud.[16]
For this project, I implemented a custom interface that communicates directly with the HTTP RESTful service. Creating a custom interface allowed me to implement calls specifically crafted for this work's needs, and therefore, to significantly reduce the number of calls to the service and speed up the whole system.

3.1.3 Stanford CoreNLP

Stanford CoreNLP provides a set of human language technology tools. CoreNLP was developed at the Stanford University in 2006, as an internal tool, and subsequently

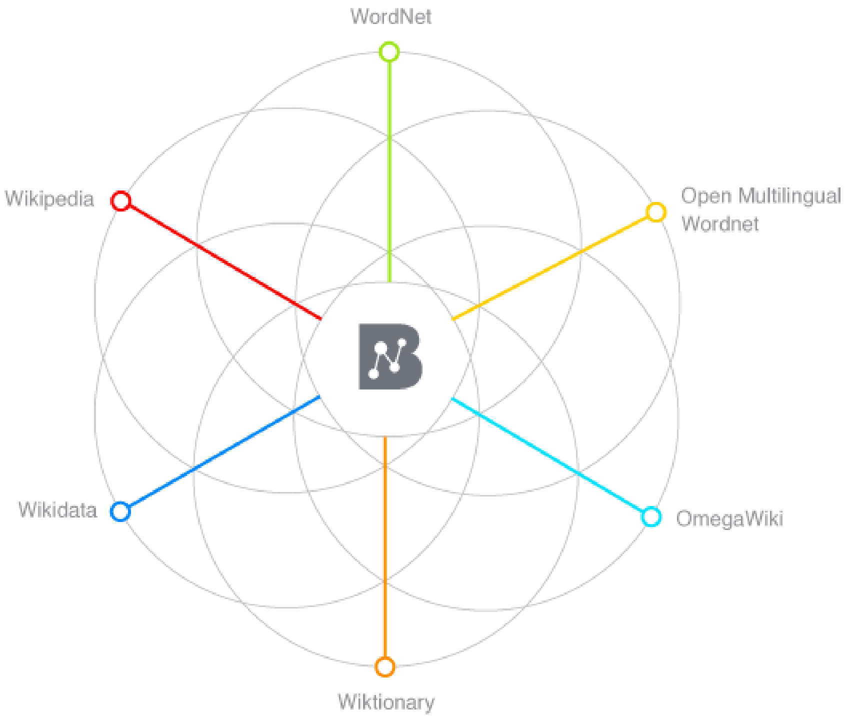


Figure 3.1: BabelNet structure overview[15]

released as a free open source software in 2010. It provides a wide variety of tools for Natural Language Processing related tasks, like giving the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalizing dates, times, and numeric quantities, marking up the structure of sentences in terms of phrases and syntactic dependencies, indicating which noun phrases refer to the same entities, indicating sentiment, extracting particular or open-class relations between entity mentions, getting quotes people said, etc.[17]
In this project, Stanford CoreNLP is used through its Java programmatic API. Adopted tools are: the part-of-speech (POS) tagger, and the Morph annotator to retrieve the Lemmas of words. What these tools do, will be better explained later on in this dissertation.

Tint NLP Tint is the first completely free and open source tool for NLP in Italian. It is based on Stanford CoreNLP, and it relies on its Java interface. Like Stanford CoreNLP, it provides an easy-to-use set of fast, accurate, and extensible Natural Language Processing modules. It was released as an open source software, under the GNU General Public License.[18]

3.1.4 Expectation Maximization

The Expectation maximization (EM) algorithm, in its general form, is an iterative method to find maximum likelihood[1] estimates of a parameter of a probability distribution, where there is a dependency on unobserved data. To better understand the problem, I will give now an example. We want to guess the current season $\theta \in$ {summer, fall, winter, spring} based on the temperature outside the window. Let's say that the probability of the temperature for each of the 24 hours of a day, is $x \in \Re^{24}$, and that we can only measure the average temperature $y = \bar{x}$ for the day. The maximum likelihood estimate of θ maximizes $p(y|\theta)$, but in some cases, this may be hard to find. This is where EM comes in help: it takes the observed data y, iteratively makes guesses about the complete data x, and iteratively finds the θ that maximizes $p(x|\theta)$ over θ.[20]

Each iteration of the EM algorithm consists of two processes: the *E-step*, and the *M-step*. In the expectation step (E-step), missing data are estimated given the observed data and current estimate of the model parameters. This is achieved using the conditional expectation, that explains the choice of terminology. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. The estimate of the missing data from the E-step is used in lieu of the actual missing data.[21]

Let's analyze the algorithm step-by-step:

Step 1: Pick an initial guess $\theta^{(m=0)}$.

Step 2: Calculate how likely it is that the complete data is exactly x, that is, calculate the conditional distribution $p(x|y, \theta^{(m)})$, where y is the observed data.

Step 3: Discharge the initial guess $\theta^{(m)}$, but keep the guess of probability of the complete data $p(x|y, \theta^{(m)})$ computed in the previous step.

Step 4: In the next step, we will make a new guess of θ that maximizes the expected $\log p(x|\theta)$. Since we do not really know x, but only a guess of the probability of distribution of x made in step 2, we will now integrate over all possible values of x, and for each one of them we will weight the probability of seeing that x, which is $\log p(x|\theta)$. However, the only thing we have is the guess made in Step 2 $p(x|y, \theta^{(m)})$.

[1] Maximum likelihood, also called the maximum likelihood method, is the procedure of finding the value of one or more parameters for a given statistic, which makes the known likelihood distribution a maximum.[19]

The expected $logp(x|\theta)$ is called the Q-function:

$$Q(\theta|\theta^{(m)}) = \text{expected} \log p(x|\theta) = E_{X|y,\theta^{(m)}}[\log p(X|\theta)] = \int_{\chi(y)} \log p(x|\theta) p(x|y,\theta^{(m)})\,\mathrm{d}x \tag{3.1}$$

where the integration is made over the support of X given y, $\chi(y)$, which is the closure of the set $\{x|p(x|y) > 0\}$.

Step 5: Make a new guess $\theta^{(m+1)}$ for θ, by choosing the one that maximizes the expected log-likelihood given in 3.1.

Step 6: Consider $m = m + 1$, and go to Step 2.[20]

In the next section, where I describe the actual implementation of this project, I will show how I applied this algorithm, which is a practical example of how the EM steps are implemented.

3.2 System structure and flow description

The system is organized in four main Scala **objects**: one, called **BabelNetApi**, to define the custom *BabelNet* API; a second, called **TintApi**, to define the custom *CoreNLP* and *Tint* APIs; a third, called **ExpectationMaximization**, that contains, in fact, the code necessary to apply *Expectation Maximization* to the retrieved data; and finally, the **WsdApi**, that runs the main code and that manages the workflow of the whole program. Beside these main objects, there are two auxiliary ones called respectively **BabelCache** and **BagCache**, that are used to manage two data structures holding data that require some time to be collected. This data is the information retrieved from BabelNet in the first case (this operation requires a bit of network time), and the *bag of words* (their meaning will be explained later on) computed with either CoreNLP or Tint, in the second.[2]

The entry point to use this service in this experimental phase, is the function *applyModifiedLeskEM*. Here is its prototype:

```
applyModifiedLeskEM(input:String, lang:Language):
                Array[Array[(Header, Double, Set[Token])]]
```

The function takes in input the *String* that needs to be processed and the language in which the sentence is written. At this stage of development, both Italian and English

[2] What follows is a description of the code. The full code can be found in appendix A

are supported, and an extension to other languages can be made with little effort. As a return value, it provides an *Array* of Arrays. Each external array represents a meaningful word in the sentence, and each internal array represents a sense for that word with an associated probability. The internal arrays are sorted by decreasing probability.
When the function starts its execution, it first converts the sentence into a **bag of word** (BOW).

Bag of Word A bag of word is a set of univocal tokens (words) in the sentence, with some associated information. In the case of this project, the associated information is:

- the **frequency** of the word in the sentence, i.e. the number of times the considered word appears in the text;
- the **lemma** of the word, that is the reduction of the word to its base (or dictionary) form. The process of *Lemmatization* uses vocabulary and morphological analysis of words to reduce inflectional forms and sometimes derivationally related forms of a word. For example, the words *run*, *runs*, *ran*, and *running* are forms of the same lexeme, with *run* as the lemma.[22] This information helps both in the grouping of the tokens in the sentence to compute their frequency, and for retrieving the appropriated information from BabelNet;
- the **Part of Speech** (POS) tag of the token. The part of speech of a word (also known as *word class* or *syntactic category*) is the relationship of this word with its neighbor words in a phrase, sentence, or paragraph. Parts of speech are mainly eight: noun, verb, pronoun, preposition, adverb, conjunction, participle and article, but the ones considered for the computed Bag of Word are only those supported by BabelNet, that are *Noun, Adjective, Verb*, and *Adverb*. POS tags are really important because they give a lot of information about a word and its neighbors, and, in WSD problems, they could completely change the meaning of a word.[5]

The second and third pieces of informations are retrieved using the Stanford CoreNLP (or Tint for Italian) annotators, specifically the *POSTaggerAnnotator* for the Part of Speech, and the *MorphaAnnotator* for the lemma.[17].
Back to the *applyModifiedLeskEM* function: for each word in the computed BOW, the program calls the function

```
getExpandedGlossBow(word:String, lang:Language,
                    pos:Option[BabelPOS] = None):
                    Map[Header, Set[Token]]
```

This function takes, as arguments, a word (in this case, the lemma of the considered word is passed), the considered language, and the POS tag. The function then returns a Map that associates to each possible sense of the treated word another Bag of Word, which this time contains all the tokens in the set of glosses for the given sense. To understand this function in detail, we first need to introduce some definitions:

Synset This is how a concept is represented in *WordNet*, and thus in *BabelNet.* It is a synonym set, i.e. the set of words that share the same meaning. The concept *wind*, for example, is expressed by the following *synset*:

$$\{\text{wind}_n^1, \text{aircurrent}_n^1, \text{currentofair}_n^1\} \qquad \text{(in WordNet notation)},$$

where the subscript of each word represents its parts of speech (*n* stands for noun), and its superscript represents the *sense* number.

Gloss A gloss is the textual definition of a *synset* (or *sense*). For example, the gloss of the above synset in WordNet is: "air moving from an area of high pressure to an area of low pressure". BabelNet, thanks to its mapping with Wikipedia, also provides a whole set of Glosses for each synset, using the definitions of all the synonym concepts aggregated in the synset.

Hypernymy and Hyponymy Given a synset S, all the synonyms in a synset H are *hypernyms* of a concept in S, if they represent a generalization of it. On the other hand, all the synonyms in synset h are *hyponyms* of a word in S, if they represent a specialization of it. For example, given balloon$_n^1$, we can consider its hypernym the set $\{\text{lighter} - \text{then} - \text{aircraft}_n^1\}$, and its hyponyms all the sets like $\{\text{hot} - \text{airballoon}_n^1\}$. [23]

Holonymy and Meronymy The meronymy relationship defines a composition or part-of relationship by relating parts to wholes. The inverse of the meronymy relationship is holonymy. Holonymy relates wholes to parts. For example *apple* is a meronym of *apple tree* and, vice versa, *apple tree* is a holonym of *apple.*[24, pp. 39]

With this concepts in mind, it is now easy to understand what happens in the function *getExpandedGlossBow*:

- First, the function retrieves all the synset ids from BabelNet that are related to the triad $(word, lang, pos)$, thanks to the custom BabelNet API:

```
val ids = BabelNetApi.synsetIds(word, lang, pos)
```

- The function then maps each synset id to a tuple containing a header with some information about the synset itself, and a Set of token representing the "extended" bag of word containing the words in the glosses of the considered synset. The fact that the bag of word is "extended", is a really important step in this work. To increase the amount of information to work with in the next step, where the actual Word Sense Disambiguation takes place, the glosses adopted are not only those of the considered synset, but also those of its hypernyms and holonyms, and of the hypernyms and holonyms of the hypernyms and holonyms, and so forth for three levels of depth. Of course, the weight of each gloss must be calibrated accordingly to the distance from the considered synset. This "extension" process takes place in the recursive function, presented next.

-
  ```
  def recGetGlossBows(depth:Int, weight:Double,
                      id:String, lang:Language):
                      List[(String, Set[tToken])]
  ```

This function operates recursively, and takes at each step the current depth, the weight of the considered synset, and the language in use. As a return value, it provides a list of tuples containing the id of each synset considered in the process and its associated bag of word. This list is then reduced back into a single set in the *getExpandedGlossBow* function. Here is what happens in the function *recGetGlossBows*:

- The aforementioned **BagCache** contains the computed bag of words containing the glosses for a given synset id. The first step this function takes, is to check if the required synset is present in the cache; if it is not, it retrieves the full synset through the custom BabelNet API, and computes the associated bag of word for all the glosses in it using the CoreNLP or Tint APIs, and puts it into the cache;
- Once it acquires the Bag of Word for the requested synset, it updates the frequency of each term in the bag, multiplying it by the *weight* passed as an argument;
- If the depth equals the maximum depth (which is set to three), the function simply returns a list containing only the tuple (synset id, bag of word); otherwise, it uses another BabelNet concept to retrieve related synsets:

Edge The concept of edges is strictly connected to the structure with which the knowledge is encoded in BabelNet. All the synsets are actually vertices V of a labeled directed graph $G = (V, E)$, where $E \subseteq V \times R \times V$

is the set of edges connecting pairs of concepts. Each edge is labeled with a *semantic relation* from R, e.g. $\{is - a, part - of, ..., \epsilon\}$, where ϵ denotes an unspecified semantic relation. A weight denoting the strength of the connection to each edge is also associated.[23]

The edges associated (connected) to a synset are retrieved using the custom BabelNet API, and they are filtered to consider only those labeled with the semantic relations *hypernym* and *holonym*;

- The last step the function takes, at this point, is a recursive call for each of the synsets connected through the edges retrieved at the previous step:

```
tBag :: edges.flatMap(e =>
          recGetGlossBows(
            depth+1,
            weigth * e.weight,
            e.target,
            lang)
        )
```

In the new call, the depth is increased by one, and the current weight is multiplied by the weight associated with the edge in use, so that the further a concept is from the main one, the lesser it impacts the disambiguation calculus.
The above code is actually the return code of *recGetGlossBows* in a non-base case of the recursion, and it shows how the function returns a List created with the bag of word it computed itself (*tBag*), concatenated with the lists returned by the computation done on all the considered edges.

- At this point, the program returned to the calling function *getExpandedGlossBow*, and the first operation that is performed is to **reduce** the list of sets of tokens returned by *recGetGlossBows* into a single set, i.e. a single Bag of Word containing all the terms found in the glosses of the considered word, plus the terms contained in the glosses of its adjacent nodes. In this step, it is important to merge together repeated tokens, in order to have a proper set without repetition. Since each token in a set of the list has a frequency that is already weighted (based on the distance between the synset of which it is part of and the synset of the token in use, and on the actual weight of the arcs connecting the two synsets), it is sufficient, at this point, to just sum the frequencies of equal terms.

- The last thing left to do, is to filter the tokens in our now merged set, in order to consider only the terms corresponding to the Part-of-speech tags that are of interest (*nouns, adjectives, verbs, adverbs*).
- The function then returns a data structure that is, as mentioned before, a map containing: as key, an header representing a particular sense of the considered word with some useful information associated, and, as value, the corresponding bag of word composed by the terms in the extended glosses of the word.

Now the computation is back to the function *applyModifiedLeskEM*. The next big step is to apply expectation maximization to the data collected so far, but before doing so, there is one more operation to perform. To explain this step, one more concept needs to be introduced:

Term frequency – Inverse document frequency (TF-IDF) This is a really important concept in information retrieval. As the term implies, TF-IDF calculates values for each word in a document, through an inverse proportion of the frequency of the word in a particular document to the percentage of documents the word appears in. Words with high TF-IDF numbers imply a strong relationship with the document they appear in, suggesting that if that word were to appear in a query, the document could be of interest to the user.[25]

The last step before applying Expectation Maximization is, in fact, computing TF-IDF for each token of each bag of word associated with each sense of each word. In the context of this project, TF-IDF is computed in the scope of all the senses of a word. This means that as the number of documents n it is considered the total number of glosses (senses) of the word to which the token refers, as the frequency f of a token it is considered the frequency previously calculated in the building of the bags, and as the document frequency df is considered the number of senses (among those considered to compute n) in which the token appears. The TF-IDF of a specific token j is then calculated with the following formula:

$$\mathrm{TF\_IDF_j} = f_j * \log \frac{n}{df_j} \tag{3.2}$$

This operation is done in the function

```
setDocumentFrequencies(
                glosses:Map[Header, Set[Token]]): Unit
```

to which is passed the Map associated to each word in the input sentence, and containing all of its possible senses.

Let's now do a recap on what data structure was obtained so far, before diving into the next phase. The process started with an input sentence that was decomposed and transformed into a set of tokens, each one with an associated POS tag and frequency. The set was then filtered to consider only a set of part-of-speeches, that is *noun, adjective, verb*, and *adverb*. For each word with its associated POS tag, then, the system retrieved all the possible senses of the word through BabelNet, and composed a bag of word considering all the terms in the glosses of each sense, plus those contained in the glosses of related concepts, and of concepts related to the related concepts, with relation types *hypernym* and *holonym*. During this process, a weight was given to each token, considering the distance from the central term, and the weight associated to the edges of the BabelNet graph. Finally, a value called TF-IDF was associated to each token, as well. The end result is and Array of Maps: each Map represents a word in the input sentence, and contains all the possible senses of that word with the associated extended bag of word.

The last step of this process is the application of Expectation Maximization, obtained through the method *compute* of the object *ExpectationMaximization*:

```
def compute(collection:Array[Map[Header, Set[Token]]]):
                Array[Array[((Header, Double), Set[Token])]]
```

This method takes as input the previously computed data structure, and returns an Array containing the Maps given in input (which contain the senses of each word), that are converted into arrays, and ordered by decreasing probability. Let's see in detail what happens in this method:

- The first step is to create a score matrix holding the scores (computed by a score function) that a particular sense of a word has, in relation with all the senses of the other words in the collection.
  Let's assume, for example, that we are considering an input sentence of four words, the first one with three possible senses, the second one with two, the third with four and the last one with only one. Let's denote the word i with w_i, and the sense j with e_j:

  $$W_0 : (e_0, e_1, e_2)$$
  $$w_1 : (e_3, e_4)$$
  $$w_2 : (e_5, e_6, e_7, e_8)$$
  $$w_3 : (e_9)$$

The score matrix will then be 10x10 with value -1 in the position (i, j), if both senses e_i and e_j belong to the same word w_k, and, if otherwise, with the computed score S_{ij}:

$$scores = \begin{pmatrix} -1 & -1 & -1 & S_{03} & S_{04} & S_{05} & S_{06} & S_{07} & S_{08} & S_{09} \\ -1 & -1 & -1 & S_{13} & S_{14} & S_{15} & S_{16} & S_{17} & S_{18} & S_{19} \\ -1 & -1 & -1 & S_{23} & S_{24} & S_{25} & S_{26} & S_{27} & S_{28} & S_{29} \\ S_{30} & S_{31} & S_{32} & -1 & -1 & S_{35} & S_{36} & S_{37} & S_{38} & S_{39} \\ S_{40} & S_{41} & S_{42} & -1 & -1 & S_{45} & S_{46} & S_{47} & S_{48} & S_{49} \\ S_{50} & S_{51} & S_{52} & S_{53} & S_{54} & -1 & -1 & -1 & -1 & S_{59} \\ S_{60} & S_{61} & S_{62} & S_{63} & S_{64} & -1 & -1 & -1 & -1 & S_{69} \\ S_{70} & S_{71} & S_{72} & S_{73} & S_{74} & -1 & -1 & -1 & -1 & S_{79} \\ S_{80} & S_{81} & S_{82} & S_{83} & S_{84} & -1 & -1 & -1 & -1 & S_{89} \\ S_{90} & S_{91} & S_{92} & S_{93} & S_{94} & S_{95} & S_{96} & S_{97} & S_{98} & -1 \end{pmatrix}$$

The score S_{ij} is actually the Lesk Score computed between the sense e_i of the word w_h and the sense e_j of the word w_k, and is computed by the function

```
def getLeskScore(
            g1:(Header, Set[Token]),
            g2:(Header, Set[Token])):Double
```

This function finds the intersection between the two sets (or bag of words) provided in input, and for each token t_i in the intersection, it computes the product of the TF-IDFs of the token i in the two sets. So, given the token i in the set 1 (t_{i1}), the token i in the set 2 (t_{i2}), the frequency of the token i in the set j (f_{ij}), and the inverse document frequency (idf_{ij}) of the the token i in the collections of senses of the word w_k of which the bag of word j is part, the function computes:

$$tf\_idf_i = f_{i1} * f_{i2} * \log(idf_{i1} * idf_{i2}) \tag{3.3}$$

After computing this value for each token in the intersection, this method simply sums all of the computed values, and returns the result as the score S_{ij} for the senses e_i and e_j.

- Once the score matrix is computed, the data is prepared for the computation with Expectation Maximization. Two values are associated with each sense: an *exp* value, representing the expected probability that is computed in the *Expectation (E)* phase of the algorithm, and the actual probability *prob*, that needs to be maximized in the *Maximization (M)* phase. The probability *prob* is, at first, set to the same value for all the possible senses e_j of a word w_k: this value is equal to $\frac{1}{n_e}$, where n_e is the number of senses of the considered word. With reference to the previous example, we have that:

$$prob_{e_0} = prob_{e_1} = prob_{e_2} = \frac{1}{3}$$

- Now, the execution of Expectation Maximization can begin. The *Expectation* phase and the *Maximization* phase are executed one after the other, until the maximum variation of probability from one iteration to the other is:

$$\Delta prob_{max} <= 0.001$$

- In the *Expectation* phase of the algorithm, the *exp* value for each sense e_i of each word w_h is computed by multiplying the probability *prob* of e_i by the value in the matrix *scores* corresponding to the Lesk Score of e_i with all the senses e_j of all the other words w_k. These values are then summed and divided by the total number of senses not in w_h. In pseudo code, considering a data structure *sensecol* containing all the senses of all the words, we have:

```
for(i in sensecol.size){
  c = 0;
  tot = 0;
  e = sensecol(i);
  for(j in sensecol.size){
    if(scores(i)(j) != -1){
      c = c + e1.prob * scores(i)(j);
      tot = tot + 1;
    }
  }
  if(tot != 0)
    e.exp = c / tot;
}
```

- The *Maximization* phase uses the *exp* value of each sense to update its probability *prob*, by summing the *exp* of each sense of a word w_k, storing the sum in a value $exptot_k$, and then computing the probability of a sense e_i by the formula:

$$prob_i = \frac{exp_i}{exptot_k}$$

If we consider the data structure *tokencol* as the collection of all the words w_k that we are considering, we have (in pseudo code):

```
for(token in tokencol){
  tot = 0;
  for(e in token){
    tot = tot + e.exp;
  }
  if(tot != 0){
    for(e in token){
      e.prob = e.exp / tot;
    }
  }
}
```

- When the iteration is finished, the method returns the collection containing all the words with all their senses - ordered by decreasing probability.

The execution is finally over, and the program can now provide all the data required by the next module of the system of which this project will be a part of, or the data that can be adopted in other use cases, as introduced earlier in this dissertation.
In this phase of implementation, the data is employed by a program used to execute a performance test for this module. The test phase will be illustrated in the next chapter.

Chapter 4

Performance evaluation

This chapter describes how the performance test was made for the project presented in the above chapters, it explains how the data set used was built, and how the evaluation was performed.

The performance test for this project was made by running the program against a set of data built to be as generic as possible; its evaluation was made by analyzing the results of some specific words, them being either nouns or verbs, in the sentences of the data set. To check the quality of the result, it was necessary to build a version of the data set called *Golden Set*, in which the meaning of the target words was hand-tagged with the most appropriate sense.
The fact that only a subset of the vocabulary of the data set was tested, does not mean that the system was disambiguating only those target words. In fact, it was working as intended - meaning that it was performing WSD on all the words of the input sentences - but only a subset of the words was checked for correctness.

4.1 Data Set

The data set used is a subset of the data used for the task 14 of the contest **SemEval 2010**[1]. For this contest, a series of 100 words was provided to be disambiguated, and along with them, a large set of training and test data, including several thousands of sentences. To generate the set, I wrote a Scala Application called *MakeDataset*, that took all the sentences provided in the contest as a training set for each of the target

[1] SemEval is a series of evaluations of computational semantic analysis systems, that started in 1998 with the name of *Senseval*, and then took the name of SemEval 20xx starting from 2007. Each edition of the contest provides a series of different tasks in the field of semantic analysis. Task 14 of SemEval 2010 was called *Word Sense Induction & Disambiguation*: participants were required to induce the senses of 100 target words, using a training set, and then disambiguate unseen instances of the same words, using the induced senses.[26]

words, as well as those provided as a test set, and filtered them to find those with the following requisites:

- contain at least **two** of the target words;
- are long at most 200 characters.

The first constraint was added because one of the most interesting parts of this work is its application in All-words tasks, so, to proper understand if the simultaneous disambiguation of different terms works properly, it was necessary to have more than one target per sentence.

The reason for the second constraint comes from the field in which this software will be employed: the conversational agents. As I mentioned before, the inputs provided in this context by users - usually - are of short length, and that is the reason why I tried to enhance the context to work with, by extending the considered glosses during the execution (Chapter 3). This explains why the capability of working with small input sentences is one of the key strengths of this project, and therefore performing the tests in this conditions was deemed necessary.

After this filtering phase, the sentences were filtered one more time by removing sentences that were too similar to other sentences present in the data set. A string $s1$ is considered to be too similar to a string $s2$, if:

```
(s1.diff(s2).length / s1.length) <= 0.10
```

where *s1.diff(s2)* is a method that returns a string containing all the character sequences contained in s1, but not contained in s2. The aim of this last filtration was to have the widest possible variety of sentences. The end result is a data set of:

- **82** target words;
- **230** sentences.

4.2 Golden Set

The creation of the *Golden Set* was completely handmade. The process started by taking each sentence in the data set - which was built as explained in the previous section - and manually searching each target word in BabelNet, to find the best suiting senses. This operation was made in collaboration with some colleagues, to cross-check the selected senses, in order to have the most objective results. Since this work would have had commercial value for the company, we wanted to make sure that the results were as reliable as possible, so that it would have a solid starting point for the implementations of the following modules of the big system (mentioned in chapter

1). The end result of this delicate operation is a Json[2] file, containing a *JsonArray* of *JsonObject*, each one of them representing a sentence in the data set. Each object has a string value that contains the sentence itself, an integer value representing the index of the sentence in the data set, and another *JsonObject* containing the target words present in the sentence, along with its associated BabelNet synset IDs. Here is an example of a fragment of the Golden Set:

```
[
  ...,
  {
    "sentence": "15.1 Any third party which has sought
      access to a gas processing facility may provide
      the information set out in rule 13.1 to the industry
      body in relation to an approach for access .",
    "index": 5,
    "tokens": {
      "body": ["bn:00011744n", "bn:00014512n"],
      "gas": ["bn:00037385n"],
      "access": ["bn:00000661n"]
    }
  },
  ...,
]
```

As shown by the fragment, sometimes the selected senses for a target word might be more than one. This happens for two reasons:

1. In some situations, the differences of meaning between two BabelNet synsets of a word, are not that evident, and usually both of them are valid in the context of the considered sentence. When this happens, all the correct ids are inserted in the set, ordered by the most fitting to the least fitting;

2. Sometimes a target word appears in the sentence, in the form of more than one part-of-speech. This means, for example, that a word can appear both as a *noun* and as a *verb*. In this case, the correct *synset ids* for all the part-of-speech are included. The discrimination between a sense referring to a part-of-speech or the other, can be done by simply looking at the synset id code.

[2] JavaScript Object Notation (JSON) is a text format for the serialization of structured data. It is derived from the object literals of JavaScript, as defined in the ECMAScript Programming Language Standard, Third Edition [ECMA].
JSON can represent four primitive types (strings, numbers, booleans, and null), and two structured types (objects and arrays).[27]

The next section will provide a thorough illustration of how both the *data set* and the *golden set* are used to run the performance test, an what the results are.

4.3 Evaluation Algorithm

The evaluation is performed in a way that takes into consideration the intended future commercial application of this software, which implies, as previously mentioned, it is meant to be a module in a larger classification system that can recognize useful concepts in user inputs; said concepts are used to enhance the experience in a conversation between a human and a chatbot, and to automatically expand the knowledge base used by the final product. This means that it is not required by this module to recognize with certainty the correct sense of a word in an input sentence; it is indeed required, though, to provide a set of most probable senses for each token - this allows subsequent modules to understand that there is a "good" probability that a certain concept is useful for the domain that the conversational agent is considering. For this reason, in the computation of the performance score, the discovery of the correct sense of a target word is taken into account, even if it is not the one with the highest probability - of course, with a penalty added to it.

This preamble leads us to the actual implementation of the performance test.
The test is run by the Scala Application *PerformanceTest*, that issues a call to the method *applyModifiedLeskEM* of the Scala object *WsdApi* for each of the sentences in the data set. When the execution for a certain sentence is complete, the application:

- Takes all the result tokens, with their associated senses and probabilities, and it checks if each one of them is among the target words considered in the golden set entry for the same sentence;
- When it finds a token corresponding to one of the target words, which will from now on be referred to as *currentToken*, it takes the array of the valid synset ids from the golden set, and filters it to consider only the part-of-speech of *currentToken*;
- Associates a temporary score, computed by normalizing the probability of the sense with the probability of the most probable one, to each one of the senses of *currentToken*, that are already sorted by decreasing probability. By doing this, we will have that the most probable sense will have a score of 1, and all the others will have a score that is inversely proportional to the distance between their probability and the probability of the most probable sense. This means

that the higher this difference, the lower the score of this sense. This way, if a sense is second, but has a probability that is really close to the one of the first, it will still have a score close to 1;

- After this score is computed, the application checks each sense of *currentToken*, and verifies if its synset id is in the array of the acceptable senses retrieved from the golden set. As mentioned in the previous section, this array is sorted from the most fitting sense to the least fitting one, so, if the synset id x is in the array, its previously computed score is multiplied by

  $$1 - (indexOf(x) * 0.1)$$

  where $indexOf(x)$ is the index of the synset id x in the array in the golden set. This means that, if the id x is the first of the list, its score will be multiplied by 1; if it is the second, it will be multiplied by 0.9; if it is the third, it will be multiplied by 0.8, and so on and so forth.
  On the other hand, if x is not in the array, its score is set to 0.

- The final score considered for *currentToken* is then set equal to the one score of its sense with the highest score computed in the previous passages.

During the execution of the test for each sentence in the dataset, the application keeps also track of:

- the number of elaborated tokens corresponding to a target word;

- the total score, computed by simply summing the scores associated with each token;

- the number of occurrences of each target word in the sentences;

- the total score for each target word.

It must be added, that the application also keeps track of the elapsed time to complete the whole test.

4.4 Results

At the end of the test, the Scala Application *PerformanceTest* computes the final accuracy percentage by dividing the total score by the number of tokens processed. It also computes an accuracy percentage for each of the target words in the same way. What follow, is the output of the performance test:

```
test completed
disambiguation performance 73.69446838540583%

Performance for each token:
failure: 100.0, with 7.0 occurrences
chip: 100.0, with 2.0 occurrences
cultivate: 100.0, with 1.0 occurrences
reputation: 100.0, with 1.0 occurrences
reap: 100.0, with 1.0 occurrences
owe: 100.0, with 1.0 occurrences
legislation: 100.0, with 1.0 occurrences
innovation: 100.0, with 1.0 occurrences
assemble: 100.0, with 1.0 occurrences
deploy: 100.0, with 1.0 occurrences
threat: 100.0, with 4.0 occurrences
shave: 100.0, with 1.0 occurrences
relax: 100.0, with 2.0 occurrences
root: 99.91920530559875, with 3.0 occurrences
mind: 96.13708699804084, with 18.0 occurrences
television: 95.29433174883343, with 10.0 occurrences
screen: 95.198794407153, with 17.0 occurrences
happen: 95.0, with 6.0 occurrences
campaign: 93.33333333333333, with 9.0 occurrences
movement: 92.65097487855236, with 6.0 occurrences
market: 91.69006351643583, with 21.0 occurrences
display: 91.2400429932824, with 25.0 occurrences
officer: 91.20968248367731, with 4.0 occurrences
signal: 90.24796446686616, with 11.0 occurrences
conclude: 90.0, with 1.0 occurrences
swim: 90.0, with 2.0 occurrences
pour: 90.0, with 1.0 occurrences
community: 89.68834382327944, with 8.0 occurrences
flight: 89.68095404227688, with 11.0 occurrences
house: 88.32707862384284, with 10.0 occurrences
divide: 87.55340603107669, with 5.0 occurrences
speed: 86.85938953778756, with 12.0 occurrences
deny: 86.59249975142203, with 5.0 occurrences
separate: 85.46752059918529, with 11.0 occurrences
frame: 85.29757937247943, with 4.0 occurrences
```

```
gas: 84.7349915245427, with 6.0 occurrences
commit: 83.67064463129506, with 1.0 occurrences
observe: 82.07112322857701, with 2.0 occurrences
promotion: 81.61273321245403, with 3.0 occurrences
cell: 80.9132159122901, with 16.0 occurrences
field: 80.22684793422353, with 18.0 occurrences
defend: 79.41992209851144, with 2.0 occurrences
air: 78.9075272077925, with 19.0 occurrences
slow: 78.60642073688308, with 5.0 occurrences
accounting: 78.48848042463555, with 9.0 occurrences
introduce: 77.37665466359498, with 4.0 occurrences
origin: 75.73075979561442, with 2.0 occurrences
camp: 73.68456112477088, with 6.0 occurrences
accommodate: 72.41038925812119, with 7.0 occurrences
mark: 70.38611050784212, with 8.0 occurrences
dealer: 70.34102938594549, with 2.0 occurrences
shape: 69.63014508833155, with 15.0 occurrences
class: 69.4941481237395, with 16.0 occurrences
idea: 66.67363832013359, with 10.0 occurrences
body: 66.34807628464357, with 35.0 occurrences
lay: 65.27773550276, with 4.0 occurrences
figure: 65.14348142301255, with 10.0 occurrences
violate: 64.92490403134586, with 2.0 occurrences
stick: 64.37983326616478, with 7.0 occurrences
margin: 62.72344159754254, with 13.0 occurrences
access: 62.060847443387814, with 28.0 occurrences
note: 59.44531231256336, with 15.0 occurrences
question: 58.350695756268, with 15.0 occurrences
foundation: 57.65710809481709, with 3.0 occurrences
gap: 57.65276162738401, with 3.0 occurrences
park: 56.82299987616678, with 10.0 occurrences
wait: 55.336679964546676, with 9.0 occurrences
road: 55.07743740298279, with 7.0 occurrences
level: 54.337528765393095, with 3.0 occurrences
lie: 53.99542652620497, with 4.0 occurrences
apply: 52.892073623562815, with 11.0 occurrences
moment: 51.21958322584076, with 13.0 occurrences
function: 51.00434825327677, with 16.0 occurrences
edge: 48.04946629300908, with 9.0 occurrences
```

```
rally: 45.10064408764317, with 3.0 occurrences
operate: 44.57782508023179, with 6.0 occurrences
office: 38.84481321272902, with 16.0 occurrences
pursue: 37.54283691593421, with 2.0 occurrences
commission: 36.412058051048604, with 2.0 occurrences
weigh: 33.593246684207834, with 4.0 occurrences
regain: 20.264578032515512, with 2.0 occurrences
expose: 17.597391364837843, with 1.0 occurrences

elapsed time 12 minutes
```

The total execution time is 12 minutes for 230 sentences. This means an average execution time per sentence of 3 seconds, which, based on the current execution times for similar applications (i.e. Watson Conversation), can be considered more than acceptable, in a human-machine interaction.
The overall accuracy is approximately 73.7%, which is also positive, considering - for example - the results obtained by the Lesk algorithm. At this stage of development, the big difference in performance among the various target words seems to be the main weakness of this system. Further research is being made to fix this issue.

This output shows all the considered target words, while the complete data set can be found in Appendix B.

Chapter 5

Conclusions

The studies made for the development of this Word Sense Disambiguation module, led to the implementation of a powerful tool able to disambiguate all the words in a short user sentence with a fairly good precision, even though the context provided is really limited. Of course, in this phase of implementation, the performances are not yet excellent, but the base idea and the system architecture provide a really good margin of improvement. Even if this project was developed for a specific use case, its structure allows an easy portability to new use cases, and to implementations with various human languages. In fact, all the languages supported by CoreNLP and BabelNet can be easily supported by this software.

The core of this work, that introduces the use of the statistical algorithm Expectation Maximization to the field of Word Sense Disambiguation, provides a new promising way of employment of the very popular Lesk algorithm on all-words tasks - without the computational complexity proper of the original implementation of the algorithm, and also without the need of having such a wide context in the sentence to apply the Simplified version of Lesk.

Bibliography

[1] R. Krovetz, "Homonymy and Polysemy in Information Retrieval," in *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, EACL '97, (Stroudsburg, PA, USA), pp. 72–79, Association for Computational Linguistics, 1997.

[2] E. Agirre and P. Edmonds, *Word Sense Disambiguation: Algorithms and Applications.* Springer Science & Business Media, Nov. 2007. Google-Books-ID: GLck75U20pAC.

[3] N. Ide and J. Véronis, "Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art," *Comput. Linguist.*, vol. 24, pp. 1–2, Mar. 1998.

[4] E. S. Raymond, "Jargon File Version 2.8.1," Mar. 1991.

[5] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, 2009. Google-Books-ID: fZmj5UNK8AQC.

[6] B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data.* Springer Science & Business Media, June 2011. Google-Books-ID: jnCi0Cq1YVkC.

[7] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220.

[8] A.-H. Tan, "Text Mining: The state of the art and the challenges," *Kent Ridge Digital Labs.*

[9] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *Proceedings of the 5th annual international conference on Systems documentation*, pp. 24–26, ACM, 1986.

[10] S. Banerjee and T. Pedersen, "An adapted Lesk algorithm for word sense disambiguation using WordNet," in *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 136–145, Springer, 2002.

[11] D. Martínez and others, “Supervised corpus-based methods for WSD,” in *Word Sense Disambiguation*, pp. 167–216, Springer, 2007.

[12] S. S. Zakirhussain, “Unsupervised Word Sense Disambiguation,” 2013.

[13] M. Odersky, S. Micheloud, N. Mihaylov, M. Schinz, E. Stenman, M. Zenger, and e. al, *An Overview of the Scala Programming Language.* 2004.

[14] “sbt Reference Manual.”

[15] “BabelNet | The largest multilingual encyclopedic dictionary and semantic network.”

[16] R. Navigli and S. P. Ponzetto, “BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network,” *Artificial Intelligence*, vol. 193, pp. 217–250, 2012.

[17] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” in *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55–60, 2014.

[18] A. Palmero Aprosio and G. Moretti, “Italy goes to Stanford: a collection of CoreNLP modules for Italian,” *ArXiv e-prints*, Sept. 2016.

[19] E. W. Weisstein, “Maximum Likelihood..”

[20] C. Yihua and R. Maya, “EM Demystified: An Expectation-Maximization Tutorial,” *Department of Electrical Engineering, University of Washington*, Feb. 2010.

[21] S. Borman, “The Expectation Maximization Algorithm A short tutorial,” June 2006.

[22] C. D. Manning, P. Raghavan, and H. Schütze, “Determining the vocabulary of terms,” in *Introduction to information retrieval*, pp. 21–33, Cambridge: Cambridge Univ. Press, reprinted ed., 2009. OCLC: 549201180.

[23] R. Navigli and S. P. Ponzetto, “BabelNet: Building a Very Large Multilingual Semantic Network,” in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL ’10, (Stroudsburg, PA, USA), pp. 216–225, Association for Computational Linguistics, 2010.

[24] L. W. Lacy, *OWL: Representing information using the web ontology language.* Trafford Publishing, 2005.

[25] J. Ramos and others, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242, pp. 133–142, 2003.

[26] S. Manandhar, I. P. Klapaftis, D. Dligach, and S. S. Pradhan, "SemEval-2010 task 14: Word sense induction & disambiguation," in *Proceedings of the 5th international workshop on semantic evaluation*, pp. 63–68, Association for Computational Linguistics, 2010.

[27] D. Crockford, "The application/json media type for javascript object notation (json)," 2006.

Appendices

Appendix A

Code

A.1 WsdApi.scala

```
package it.reply.szuccato.tesinlp.wsd

import com.typesafe.scalalogging.Logger
import it.reply.szuccato.tesinlp.babelnet._
import it.reply.szuccato.tesinlp.tint.TintApi
import it.reply.szuccato.tesinlp.tint.types.tToken
import it.reply.szuccato.tesinlp.wsd.types.{Header, Token}


object WsdApi {
  val maxDepth = 3
  val logger = Logger("wsdLogger")


  def filterToken(tokens: Set[tToken]): Set[Token] = {
    val newTokens = tokens.map(t =>
      new Token(t.text,
        t.pos match {
          case "S" => NOUN
          case "A" => ADJECTIVE
          case "V" => VERB
          case "B" => ADVERB
          case _ => null
        },
        t.lemma, t.frequency, 1, 1, 0)
    )
    newTokens.filter(_.pos != null)
  }

  def filterEngToken(tokens: Set[tToken]): Set[Token] = {
```

```
  val newTokens = tokens.map(t =>
    new Token(t.text,
      t.pos match {
        case "NN" | "NNS" | "NNP" | "NNPS" => NOUN
        case "JJ" | "JJR" | "JJS" => ADJECTIVE
        case "VB" | "VBD" | "VBG" | "VBN" | "VBP" | "VBZ" =>
          VERB
        case "RB" | "RBR" | "RBS" => ADVERB
        case _ => null
      },
      t.lemma, t.frequency, 1, 1, 0)
  )
  newTokens.filter(_.pos != null)
}

def setDocumentFrequencies(glosses: Map[Header, Set[Token]]):
Unit = {
  for (bag <- glosses) {
    bag._1.nDocs = glosses.size
    for (token <- bag._2) {
      for (bag2 <- glosses)
        if (bag._1.id.id != bag2._1.id.id)
          for (token2 <- bag2._2)
            if (token.equals(token2))
              token.docFrequency += 1
      token.tfIdf = token.frequency * scala.math.log(
        bag._1.nDocs / token.docFrequency)
      token.idf = bag._1.nDocs / token.docFrequency
    }
  }
}

private def recGetGlossBows(depth: Int, weigth: Double,
                            id: String, lang: Language):
List[(String, Set[tToken])] = {
  logger.info(s"${
    (0 to depth).map(n => "\t").
      reduce(_ + "" + _)
  }level $depth " +
    s"- retrieving glosses of synset <$id> in language " +
    s"<$lang>")

  val tBag: (String, Set[tToken]) = (id, {
    val bag = BagCache.get(s"$id-${lang.get}")
    if (bag == null) {
      logger.info(s"BagCache miss retrieving BOW for synset " +
```

```scala
        s"$id in lang ${lang.get}")
      val syn = BabelNetApi.synset(id, lang)
      val glosses = syn.glosses.filter(_.language == lang
        .get).map(
        _.gloss).reduceOption(_ + ". " + _).getOrElse("")
      val bag2: Set[tToken] = if (glosses.nonEmpty) {
        if (lang.get == "IT") TintApi.getBagOfWord(glosses)
        else TintApi.getEngBagOfWord(glosses)
      } else Set()
      BagCache.put(s"$id-${lang.get}", bag2)
      bag2
    } else
      bag
  })

  if (tBag._2.nonEmpty) tBag._2.foreach(_.frequency *= weigth)
  if (depth < maxDepth) {
    val edges = BabelNetApi.edges(id).filter(e => e.weight >
      0.0 && (
      e.pointer.relationGroup.toLowerCase.indexOf
      ("hypernym") > -1 ||
        e.pointer.relationGroup.toLowerCase.indexOf
        ("holonym") > -1
      ))
    tBag :: edges.flatMap(e => recGetGlossBows(
      depth + 1, weigth * e.weight, e.target, lang))
  } else List(tBag)
}

def getExpandedGlossBow(word: String, lang: Language, pos:
Option[BabelPOS] = None):
Map[Header, Set[Token]] = {
  logger.info(s"getting ids for word <$word> in language " +
    s"<${lang.get}>")
  val ids = BabelNetApi.synsetIds(word, lang, pos)
  val tbags = ids.map(id => {
    logger.info(s"considering synset ${id.id}")
    (new Header(word, id, BabelNetApi.synset(id.id, lang),
      0), {
      val bow = recGetGlossBows(0, 1, id.id, lang).groupBy(
        _._1).map(_._2.map(_._2)).map(e => e.sortWith(
        _.map(_.frequency).sum > _.map(_.frequency).sum)
        .head).toList
      bow.reduce((x, y) => {
        x.intersect(y).map(t => {
          t.frequency += y.find(_ == t).get.frequency;
```

```
            t
          }) | x | y
        })
      })
    }).filter(_._2.nonEmpty)
    val bags =
      if (lang.get == "IT") tbags.map(t => (t._1, WsdApi
        .filterToken(t._2)))
      else tbags.map(t => (t._1, WsdApi.filterEngToken(t._2)))
    bags.foreach(elem => logger.info(
      "SynsetId: " + elem._1.id.id + ", BOW: " + elem._2.map(
        t => s"(${t.lemma}, ${t.frequency})").reduceOption(_ +
        _).getOrElse("")))
    bags.toMap
  }

  def getBOW(sentence: String): Set[Token] = {
    filterToken(TintApi.getBagOfWord(sentence))
  }

  def getEngBOW(sentence: String): Set[Token] = {
    filterEngToken(TintApi.getEngBagOfWord(sentence))
  }

  def applyModifiedLeskEM(input: String, lang: Language):
  Array[Array[(Header, Double, Set[Token])]] = {

    val inputWords =
      if (lang.get == "EN") getEngBOW(input).toArray
      else getBOW(input).toArray

    logger.info("Getting gloss bags of the sentence: " + input)
    val windowGlsBags = inputWords.map((w) =>
      getExpandedGlossBow(w.lemma, lang, Option(w.pos)))
      .filter((g) =>
        g.nonEmpty)

    windowGlsBags.foreach(setDocumentFrequencies)

    // Compute best solution
    logger.info("Launching Expecation Maximization on " +
      windowGlsBags.length + " concepts")
    val solution = ExpectationMaximization.compute(windowGlsBags)

    solution.map(_.map(e => (e._1._1, e._1._2, e._2)))
  }
```

```
}
```

A.2 ExpectationMaximization.scala

```
package it.reply.szuccato.tesinlp.wsd

import com.typesafe.scalalogging.Logger
import it.reply.szuccato.tesinlp.wsd.types.{Header, Token}


object ExpectationMaximization {
  val logger = Logger("wsdLogger")

  def getLeskScore(g1: (Header, Set[Token]), g2: (Header,
    Set[Token])): Double = {
    val inters = g1._2.map(t => (t, g2._2.find(_.equals(t))
      .orNull)).filter(_._2 != null)
    inters.map(e => (e._1.frequency * e._2.frequency) * math
      .log(e._1.idf + e._2.idf)).sum
  }

  def compute(collection: Array[Map[Header, Set[Token]]]):
  Array[Array[((Header, Double), Set[Token])]] = {
    val as_indexed_coll = collection.map(_.toIndexedSeq)
    val pre_scores = as_indexed_coll.flatMap(m => m.map(e =>
      (as_indexed_coll.indexOf(m), e)))
    val triScores = pre_scores.map(e1 => pre_scores.map(e2 =>
      if (pre_scores.indexOf(e1) <= pre_scores.indexOf(e2)) {
      if (e1._1 == e2._1) -1.0 else getLeskScore(e1._2, e2._2)
    } else 0))
    val scores = triScores.indices.map(i => triScores(i)
      .indices.map(j => if (i > j) triScores(j)(i) else
      triScores(i)(j)).toArray).toArray
    val prob_coll = as_indexed_coll.map(elem => elem.map(sense
    => ((sense._1, new {
      var exp: Double = 0.0;
      var prob: Double = 1 / elem.length.asInstanceOf[Double]
    }), sense._2)))

    var deltaProb = 1.0
    while (deltaProb > 0.001) {
      deltaProb = 0.0
      var oindex = 0
      for (token <- prob_coll) {
        for (sense <- token) {
          var c, tot = 0.0
```

```scala
          var iindex = 0
          for (token2 <- prob_coll) {
            for (sense2 <- token2) {
              if (scores(oindex)(iindex) != -1) {
                c += (sense2._1._2.prob * scores(oindex)(iindex))
                tot += 1.0
              }
              iindex += 1
            }

          }
          if (tot != 0) sense._1._2.exp = c / tot
          oindex += 1
        }

      }

      for (token <- prob_coll) {
        val tot = token.map(_._1._2.exp).sum
        if (tot != 0) {
          for (sense <- token) {
            val preProb = sense._1._2.prob
            sense._1._2.prob = sense._1._2.exp / tot
            val dProb = math.abs(sense._1._2.prob - preProb)
            if (dProb > deltaProb) deltaProb = dProb
          }
        }
      }
    }

    prob_coll.map(t => t.sortWith(_._1._2.prob > _._1._2.prob)
    ).map(t => t.map(s => ((s._1._1, s._1._2.prob), s._2)).toArray)
  }
}
```

A.3 TintApi.scala

```scala
package it.reply.szuccato.tesinlp.tint

import java.io.File
import java.util.Properties

import edu.stanford.nlp.ling.CoreAnnotations
.{PartOfSpeechAnnotation, SentencesAnnotation, TokensAnnotation}
import edu.stanford.nlp.pipeline.{Annotation, StanfordCoreNLP}
import eu.fbk.dh.tint.runner.TintPipeline
import it.reply.szuccato.tesinlp.tint.types.tToken

import scala.collection.mutable
import scala.collection.JavaConverters._


object TintApi {
  val tintPipeline: TintPipeline = {
    val pipeline = new TintPipeline()
    pipeline.loadPropertiesFromFile(new File("resources/tint" +
      ".properties"))
    pipeline.load()
    pipeline
  }

  val corePipeline: StanfordCoreNLP = {
    val props = new Properties()
    props.setProperty("annotators", "tokenize, ssplit, pos, " +
      "lemma")
    new StanfordCoreNLP(props)
  }

  def getEngBagOfWord(text: String): Set[tToken] = {
    var bag: mutable.Set[tToken] = mutable.Set()
    val document = new Annotation(text)
    corePipeline.annotate(document)
    val sentences = document.get(classOf[SentencesAnnotation])
      .asScala.toArray

    for (sentence <- sentences) {
      for (token <- sentence.get(classOf[TokensAnnotation])
        .asScala) {
        val nToken = new tToken(token.originalText(), token
          .get(classOf[PartOfSpeechAnnotation]),
          token.lemma(), 1)

        val optOToken = bag.find(_.equals(nToken))
        if (optOToken.isDefined) {
          val oToken = optOToken.get
          bag -= oToken
          oToken.frequency += 1.0
          bag += oToken
        }
        else
          bag.add(nToken)
      }
    }
    bag.toSet

  }

  def getBagOfWord(text: String): Set[tToken] = {
    var bag: mutable.Set[tToken] = mutable.Set()
    val annotation = tintPipeline.runRaw(text)

    val sentences = annotation.get(
      classOf[SentencesAnnotation]).asScala
    for (sentence <- sentences) {
      for (token <- sentence.get(
        classOf[TokensAnnotation]).asScala) {
        val nToken = new tToken(
          token.originalText(), token.get(
            classOf[PartOfSpeechAnnotation]), token.lemma(), 1)
        val optOToken = bag.find(_.equals(nToken))
        if (optOToken.isDefined) {
          val oToken = optOToken.get
          bag -= oToken
          oToken.frequency += 1.0
          bag += oToken
        }
        else
          bag.add(nToken)
      }
    }

    bag.toSet
  }
}
```

A.4 BabelNetApi.scala

```scala
package it.reply.szuccato.tesinlp.babelnet


import com.typesafe.scalalogging.Logger
import it.reply.szuccato.tesinlp.babelnet.types._
import it.reply.szuccato.tesinlp.wsd.BabelCache
import org.json.{JSONArray, JSONObject}

import scalaj.http.{Http, HttpResponse}


```

```scala
object BabelNetApi {
  val key = "< babelNet key here >"
  val connTimeoutMs = 120000
  val readTimeoutMs = 120000

  val logger = Logger("wsdLogger")

  def version: String = {
    val response: HttpResponse[String] = Http(
      "https://babelnet.io/v4/getVersion").param("key", key)
      .timeout(connTimeoutMs, readTimeoutMs).asString
    val json = new JSONObject(response.body)
    json.getString("version")
  }

  def synsetIds(word: String, lang: Language, pos:
  Option[BabelPOS]): List[SynsetId] = {
    val jsonA = try {
      new JSONArray(BabelCache.get(s"ids-$word-${
        lang.get
      }-${pos.getOrElse("none")}"))
    } catch {
      case _: Exception =>
        logger.info(s"\tBabelCache miss retrieving ids for " +
          s"$word")
        var pMap = Map[String, String]("word" -> word, "langs"
          -> lang.get, "filterLangs" -> lang.get, "key" -> key)
        if (pos.isDefined)
          pMap += "pos" -> pos.get.get
        val response: HttpResponse[String] = Http(
          "https://babelnet.io/v4/getSynsetIds").params(pMap)
          .timeout(connTimeoutMs, readTimeoutMs).asString
        val babelJson = new JSONArray(response.body)
        BabelCache.put(s"ids-$word-${lang.get}-${
          pos.getOrElse("none")
        }", babelJson.toString)
        babelJson
    }

    val list = for (i <- 0 until jsonA.length()) yield {
      SynsetId(jsonA.getJSONObject(i))
    }
    list.toList
  }

  def synset(id: String, filterLangs: Language = IT): Synset = {
    val json = try new JSONObject(BabelCache.get
    (s"synset-$id-${filterLangs.get}")) catch {
      case e: Exception =>
        logger.info(s"\tBabelCache miss retrieving synset $id")
        val response: HttpResponse[String] = Http(
          "https://babelnet.io/v4/getSynset").params(("id", id),
          ("filterLangs", filterLangs.get), ("key", key))
          .timeout(connTimeoutMs, readTimeoutMs).asString
        val babelJson = new JSONObject(response.body)
        BabelCache.put(s"synset-$id-${filterLangs.get}",
          babelJson.toString)
        babelJson
    }
    Synset(json)
  }

  def senses(word: String, lang: Language, pos:
  Option[BabelPOS] = None): List[Sense] = {
    val jsonA = try new JSONArray(BabelCache.get
    (s"senses-$word-${lang.get}-${pos.getOrElse("none")}"))
    catch {
      case e: Exception =>
        logger.info(s"\tBabelCache miss retrieving senses for " +
          s"word $word")
        var pMap = Map[String, String]("word" -> word, "lang"
          -> lang.get, "filterLangs" -> lang.get, "key" -> key)
        if (pos.isDefined)
          pMap += "pos" -> pos.get.get
        val response: HttpResponse[String] = Http(
          "https://babelnet.io/v4/getSenses").params(pMap)
          .timeout(connTimeoutMs, readTimeoutMs).asString
        val babelJson = new JSONArray(response.body)
        BabelCache.put(s"senses-$word-${lang.get}-${
          pos.getOrElse("none")
        }", babelJson.toString)
        babelJson
    }
    val list = for (i <- 0 until jsonA.length()) yield {
      Sense(jsonA.getJSONObject(i))
    }
    list.toList
  }

  def edges(id: String): List[Edge] = {
    val jsonA = try new JSONArray(BabelCache.get(s"edges-$id")
    ) catch {
```

```scala
      case e: Exception =>
        logger.info(s"\tBabelCache miss retrieving edges for " +
          s"synset $id")
        val response: HttpResponse[String] = Http(
          "https://babelnet.io/v4/getEdges").
          params(("id", id), ("key", key)).
          timeout(connTimeoutMs, readTimeoutMs)
          .asString
        val babelJson = new JSONArray(response.body)
        BabelCache.put(s"edges-$id", babelJson.toString)
        babelJson
    }
    val list = for (i <- 0 until jsonA.length()) yield {
      Edge(jsonA.getJSONObject(i))
    }
    list.toList
  }

  def glosses(word: String, lang: Language, pos:
  Option[BabelPOS] = None): Map[(SynsetId, Synset), String] = {
    logger.debug(s"getting glosses for concept <$word> for " +
      s"language <$lang> and pos <${pos.get}>")
    val ids = synsetIds(word, lang, pos)
    val glosses = ids.map(id => (
      id, synset(id.id, lang))).map(h => (h, h._2.glosses.filter(
      _.language == lang.get)))
    val glossString = glosses.map(gl => (gl._1, gl._2.map(g =>
      g.gloss).reduceOption(_ + ". " + _).getOrElse("")))
      .filter(_._2.nonEmpty)
    logger.debug(s"\tfound ${glossString.length} senses for " +
      s"concept $word")
    glossString.foreach(g => logger.debug(s"\t\tsynsetId: ${
      g._1._1.id
    }, gloss: ${g._2}"))
    glossString.toMap
  }
}

sealed trait Language {
  def get: String
}

case object IT extends Language {
  def get = "IT"
}

```

```scala
case object EN extends Language {
  def get = "EN"
}

sealed trait BabelPOS {
  def get: String
}

case object NOUN extends BabelPOS {
  def get = "NOUN"
}

case object ADJECTIVE extends BabelPOS {
  def get = "ADJECTIVE"
}

case object VERB extends BabelPOS {
  def get = "VERB"
}

case object ADVERB extends BabelPOS {
  def get = "ADVERB"
}
```

A.5 PerformanceTest.scala

```scala
import java.time.Instant

import com.typesafe.scalalogging.Logger
import it.reply.szuccato.tesinlp.babelnet.EN
import it.reply.szuccato.tesinlp.wsd.WsdApi
import org.json.JSONArray

import scala.io.Source


object PerformanceTest extends App {
  def truncateAt(n: Double, p: Int): Double = {
    val s = math pow(10, p)
    math.rint(n * s) / s
  }

  val currTime = System.currentTimeMillis()

  val logger = Logger("test")

  val goldenSet = new JSONArray(Source.fromFile
  ("resources/goldenSet.json").mkString)
  val tokens = Source.fromFile("resources/dataset/tokens.txt")
    .getLines().map((_, new {
    var score = 0.0;
    var num = 0.0
  })).toMap


  val destFileName = s"performanceTestOut_${
    Instant.now.getEpochSecond
  }.json"
  val source = Source.fromFile("resources/dataset/dataset.txt")
  val lines = try source.getLines().toList finally source.close()
  val lang = EN
  var totCount = 0.0
  var totScore = 0.0
  for (line <- lines) {
    val index = lines.indexOf(line)
    val sentence = line
    logger.info(s"processing string $index: '$sentence'")
    val res = WsdApi.applyModifiedLeskEM(sentence, lang)
    for (elem <- res) {
      val curToken = elem(0)._1.word
      if (goldenSet.getJSONObject(index).getJSONObject
      ("tokens").has(elem(0)._1.word)) {
        logger.info(s"\tconsidering token '$curToken':")
        val pos = elem(0)._1.id.pos match {
          case "NOUN" => 'n';
          case "VERB" => 'v';
          case "ADJECTIVE" => 'a'
        }
        val expected = goldenSet.getJSONObject(index)
          .getJSONObject("tokens").getJSONArray(elem(0)._1
          .word).toList.toArray(Array[String]()).filter(
          _.last == pos)
        val scores = elem.map(e => (e._1.id.id, e._2 / elem(0)
          ._2)).map(e => if (expected.contains(e._1)) (e._1, e
          ._2 * (1 - expected.indexOf(e._1) * 0.1)) else (
          e._1, 0.0))
        val max = scores.max(Ordering.by[(String, Double),
          Double](_._2))
        logger.info(s"\t\tthe correct synset is '${max._1}', " +
          s"with a score of ${max._2}")
        totScore += max._2
        tokens(curToken).score += max._2
        totCount += 1
        tokens(curToken).num += 1
      }
    }

  }
  val orderedTokens = tokens.toArray.sortWith((t1, t2) => (
    t1._2.score / t1._2.num) > (t2._2.score / t2._2.num))

  val percentage = (totScore / totCount) * 100
  logger.info("")
  logger.info("test completed")
  logger.info(s"disambiguation performance $percentage%")
  logger.info("")
  logger.info("Performance for each token:")
  for (token <- orderedTokens) logger.info(s"${token._1}: " +
    s"${token._2.score / token._2.num * 100}, with ${
      token._2.num
    } occurrences")
  logger.info("")
  logger.info(s"elapsed time ${
    ((System.currentTimeMillis() -
      currTime) / 1000) / 60
  } minutes")
}
```

Appendix B

Data Set

7. The memory access system of claim 6 wherein the Done signal generated by the memory access logic is coupled to the queue to accommodate shifting the queue to the next queued request . Not only can members make and air unique non-commercial programming , they elect and serve as our community board of directors and ensure public access media grows in Prescott. Join Now
Direct memory access encodes the memory address in the instruction that to reads or writes the memory. With direct memory accesses , you may access variables in the memory class DATA and BIT .
6. Ofcom is also empowered to exclude certain types of programme or service from the requirement to provide television access services , or apply different targets to excluded programmes .
15.1 Any third party which has sought access to a gas processing facility may provide the information set out in rule 13.1 to the industry body in relation to an approach for access .
Double action fingerprint and access code security door lock for your office , business or home. Get 100 security at your entry point with this presidential style deadbolt double action lock .
Houghton ’s front door is accessible via a short flight of steps up from the level of Quincy Street. For wheelchair access , please contact Public Services at 617-495-2440 to make an appointment .
Similarly an employee who lost an ID card need not worry about someone gaining access using his card. The reader can be programmed to deny entry and even raise an alarm should that card be used .

Biometric fingerprint security door lock for your office , business or home. This is the highest grade of factory-direct fingerprint access entry system on the market today !
One handicapped parking space and handicapped-access entry available at the back of McGlothlin-Street Hall. Use the service access road off of Landrum Drive. Elevator access to all floors .
Cox is upgrading the security for access to the View/Pay Bill section of the website. To accommodate this upgrade , you may need to change your current sign-in email address and password .
Khoury 's has an area in the restaurant that can accommodate large parties and is separate from the rest of the dining room. We also have the patio that can be reserved for private function .
Hemingway Software Products - Produces accounting and office skill testing programs for data entry , typing , math , spelling and grammar. Behavioral assessments are also available .
People are not asking for mark to market accounting now for manufacturing and service companies; only perhaps for financial companies.
reconciling accounting records to ASAP accounts. The system produces various reports for field office and recipient use. For further information , see I TFM 6 .
Smith , A. and T. Dyckman.1981. The impact of accounting regulation on the stock market : The case of oil and gas companies : A comment. The Accounting Review (October) : 959-966. (JSTOR link) .
If you went to mark to market accounting [it could possibly solve problems related to business combinations] .
Deloitte has closed the gap on market leader PricewaterhouseCoopers to $ 71million as the largest accounting network in the world. IAB predicts it will push past PwC in 2010
1st regular journal entry to record cash transaction occurs first , during the accounting period (to record the purchase of fixed asset e.g. office equipment for $ 10 , 000) :
A digital cordless handset with a crisp color display works like a normal cordless phone. It also provides interactive access to a shared address book , call logs and voicemail .
The process starts by the recursor asking one of these root servers - for example , the server with the IP address " 198.41.0.4 " - the question " what is the IP address for www.wikipedia.org ? "

We admit that every Convict in New South Wales is bound , in common gratitude , to sign an address to Mr Forbes ... but ... the body of respectable Emigrant Colonists owe him no such obligation .
Free speed 320 RPM; 35 ft.-lbs. torque 82 dB sound level 3.5 CFM 90 PSI operating pressure 1/4in. air inlet , Manufacturer : AirCat
Smoke stick : A diagnostic tool used to observe air flow. Usually it consists of a chemical in a squeezable container. When squeezed it emits smoke which visibly follows air flow currents .
Because getting fresh air is great for your body and mind ... while indoor air is quite possibly hazardous for your health. According to the U.S. Environmental Protection Agency (EPA) :
a device which adds humidity , or moisture , to the air. Humidifiers typically operate by passing the house air through an evaporative surface that is constantly replenished by a water source .
The speed of sound is n't easy to define - it varies depending on air conditions and altitude. At sea level it is approximately 760mph , while at 30 , 000ft it 's around 660mph .
CUT-OFF LOW : An upper level low pressure system that is no longer in the normal west to east air flow. Usually a cut-off low will lie to the South of the established upper air flow .
Improving the air quality in your home is also a good idea. You can do this by using an air cleaner , venting all gas appliances to the outside , and avoiding wood fires in your house .
705.104 (1) Subject to subsection (3) , no air operator shall operate an aircraft with passengers on board unless the crew includes at least the following number of flight attendants :
To apply for this job , fill in the information below then click on " Send " . Enter your e-mail address carefully. The Employer will not be able to respond to you if this field is incorrect .
For the lips you have the lip liner that is used to give a perfect shape to the lips. Along with that you have lip brushes that are used to apply lip stick , lip gloss and lip conditioner .
If a client will wait for a 100 (Continue) response before sending the request body , it MUST send an Expect request-header field (Section 14.20) with the " 100-continue " expectation .

What if everything You thought You knew about body armor was a lie ? Click Here For FREE access to the Test Results. This is a life or death advance in bullet proof technology ! clicking here .
This subclass is indented under subclass 146.1. Vehicle body access barrier or closure including manually engaged means at a separate location from that of the barrier or closure which
2.5. The information stand in the offices of an Antimonopoly body (in its community liaison office or at the entrance) must display the following information :
White blood cell White blood cells (also called leukocytes or immune cells) are cells which form a component of the blood. They help to defend the body against ... > read more
[0041] FIG. 3 is an enlarged view showing one hexagonal shaped cell 3a having an perfect hexagonal shape in the hexagonal cell honeycomb structure body 1 according to the first embodiment .
1699 , " quantity of motion of a moving body , " from L. momentum " movement , moving power " (see moment) . Fig. use dates from 1782 .
5 Wheatgrass is high in oxygen like all green plants that contain chlorophyll. The brain and all body tissues function at an optimal level in a highly-oxygenated environment .
29. The article of claim 25 , wherein the body of the incoming electronic message comprises a target address field including an address of an intended recipient of the incoming electronic message .
" I would know because my body shape would change , " you answer. " If I lost body fat , I would be thinner , my clothes would be looser , and I would weigh less. "
where F is the force acting on the body and a is the acceleration of the body. [note 2] For the moment , we will put aside the question of what " force acting on the body " actually means .
II. Second , even if a woman 's body is her own property (which no libertarian would deny) , the question still remains whether the body of the unborn child is also the woman 's property .
A method of measuring your body fat in which you lie on your back while a signal travels from an electrode on your foot to an electrode on your hand. The slower the signal , the more fat you have .

Would you mind posting a note on the web site for a open invite for above dates. Any and all good guys and sore heads swim , ski , lunch etc , etc ... David Bruce , THE MAYOR OF CAMP PARADOX .
Pench Jungle camp offer 12 deluxe safari style canvas tents housed on permanent foundation with equipped with showers and tiled bathrooms. Each spacious tent can easily accommodate an extra bed .
James Irvin Civic Center , 38122 Martin Luther King Blvd. , Dade City , can accommodate 40 children; no cost for this camp , plus field trips. Camp runs 8 : 30 a.m. to 3 : 30 p.m .
Newsweek columnist Jonathan Alter , traveling with the Obama campaign , said the attempt by the Clinton camp to weigh various stories represented a kind of " silly , even-Steven-itis. "
What do you think is going to happen ? People are put into those camps for safety. LOL Anyone put into a camp is labeled a threat in one way or another. So think about it and look at your children .
By talking to Texas television stations , the Obama camp learned Mrs. Clinton ’s campaign had prepared a commercial to be shown at 7 a.m. on Friday , although no one knew the contents .
Portman faces a primary challenge from Northeast Ohio car dealer Tom Ganley , who officially launched his campaign July 1. Ganley raised $ 103 , 600 , campaign spokesman Jeff Longstreth said .
The use of quick responses and television access to control , shape and change voters ’ minds on a daily basis makes this the most technologically evolved Senate campaign ever. And the costliest .
On the fourth floor of 175 Highland Ave. , state Senate candidate Sara Orozco and volunteers huddled together in a small office room , using cell phones to call folks about Orozco ’ s campaign .
After the campaign he became communications director for Sen. Harry Reid (D-Nev.) . The idea for Soapbox evolved as Naylor recognized the niche market in servicing on-the-road journalists .
A protein formed by the body to fight infections. Antibodies bind to antigens (cell markers) on viruses , bacteria , or tumor cells and mark them for destruction by the body ’ s immune system .
A cell marker produced by all cell types. Antigens help the body ’ s immune system identify cells. The immune system uses antibodies to bind to the antigens in order to mark the cells for removal .

Refusal of prison to allow inmate to keep memory typewriter and carbon paper in cell did not violate his right of access to courts. Sands v. Lewis , 878 F.2d 1188 (9th Cir. 1989) .
Here ' s a partial list of studies describing compromised function and/or overt disease assoctiated with exposure to cell phones. The list includes lay descritions whenever possible :
Seizure of legal materials from inmate 's cell and holding of them for three days did not violate right of access to courts. Vigliotto v. Terry , 865 F.2d 1131 (9th Cir. 1989) .
Contents : The idea of enhancing the immune system 's ability to cope with HIV has ... Every cell in our body (except for red blood cells) contains a complete copy of our DNA ...
Paralyzed rats are walking again , thanks to a new stem cell treatment; that 's big news for the medical community ... and a 10-year wait for anybody wanting to make a buck off it .
Use standard playing cards with your students to introduce cellular interactions such as cell to cell recognition and signal and receptor specificity. Preview Download Print-and Go (PDF)
As you can see , we 've added a new challenge to Mr. Ma. Let 's hope he views this as a new opportunity. We 're ready to join with him in a campaign aimed to slow consumption of shark fin soup .
As Japan attempts to weather the global financial turmoil , experts say it ' s biggest challenge is not the shape of the banks or stock market but the country ' s aging population. (AFP TV World)
Clock signal to DoCDTM On chip Debugger module. This separate clock line allow DoCDTM to operate during the SLEEP mode (major clock CLK is stopped) .
The memory devices connect directly to the system buses. Because each device is only 32K , each uses only 15 address lines. Notice how each device is activated by a separate chip select .
I am of the slow thinking laboring class , the 70 of society with below collage level intelligence. We who have no appreciation for classical music , we who lack the brains to understand it .
Rowing - Row , row , row your body to a new cardio level with this low impact 30 minute class on our rowing machines. Sign-up is required for this class. (30 minutes , 4 week class)

To register , just click the Quick Register " icon , enter your first name and last name in the body field and click send ! Your name will be added to the listing for that specific group class !
Bear in mind that blue-collar workers can also commit so-called white-collar crime. Blue-collar workers have access to non-cash assets , such as supplies , tools , inventory , and equipment .
the kid has no idea how much he does not understand. perhaps he shopuld stick to community organizing. Posted by Jim Johnson August 28 , 08 08 : 50 PM
Not only to improve our member ’ s skill and technical knowledge , but also to cultivate financial knowledge , legal mind and accounting discipline. You are expected to broaden your knowledge .
But would I deny the dealer access to our church ? No , he certainly needs to be and so does the young man whose life has been devastated. Should we be more proactive to watch for this , certainly .
7.1.4 Suspend or deny access to all or any part of any email or other communication address issued to You by notice if we believe or receive a complaint related to the content or the use thereof .
Figure 3a. A digital radiograph of the knee taken with a high resolution CR screen (HR screen , Eastman Kodak Company) is illustrated using display processing with no edge restoration .
NOTE : If applicable foro their class and display , exhibitors must bring their own computers for demonstration purposes; computers WILL NOT be provided. Internet access will not be available .
With support for up to 150 hours of video and a dazzling 2.5-inch color display , the new iPod video lets you take music videos , TV shows , movies and video podcasts on the road or in the air .
Output screen - Device used to display output. This could be a separate monitor or other display device used only to display the output being received from the computer or other device .
Once the two new nuclei are formed , the chromosomes de-condense and gene transcription can resume (i.e. genes can begin to function) . All that remains is for the cell to divide in two .
We use the oncogene sis as an example. sis encodes a protein that is almost the same as a growth factor named PDGF. The cell receiving the fake signal will divide wrongly , resulting in a tumour .

People of any skill level/bike are welcome (road bike is recommended , but not limited to) . We divide up into few groups if needed , to accommodate for the difference in skill levels .
" There ’s no one that people want to rally behind yet , " the Communication junior said. " But also , there are n’t really enough people to divide into separate groups for each candidate. "
31.08.2006 to cinema , edge , hollywood , market , marketing , movie , promotion , snowball , soap , strategy , success , user generated by bertram
... or space forming a boundary : border , borderline , brim , brink , edge , edging , fringe , margin , periphery , rim , verge. See EDGE. 2. A line around a closed figure or area : ...
Touch the razor edge lightly and note the reaction. A dull edge produces no drawing feeling. A razor that has the proper cutting edge tends to stick to the thumb and will not slide along it .
A big button amplified telephone with adjustable volume , hands-free function and dedicated 1571 button to pick up BT ’s 1571 answer service. An easy to use entry level phone .
The " MainClass " entry may be appended by the parameters expected by the " main ( ) " method when invoked. Caution : mind the case. Java class names are case sensitive !
Short Entity A complete match. The Name Control may be replaced by the Check Digit and the system will display : SHORT ENTITY. The cursor will stop in the next field for entry .
Click OK to return to the edit page. Then type a unique entry in the field. You can see a list of entries in the field by clicking the Word Wheel or Browse Choices button on an HTML query screen .
Reply : Freelance Data Entry worker. I am a data entry worker with an experience of 3 years experience in the field and a very good typing speed of 75 wpm .
We drop the water level down just enough to expose the cleaning area. After cleaning , we apply a sealant coating to the tile to richen the original color and luster. LOOKS JUST LIKE NEW !
Kidney Disease Kidneys are particularly prone to damage from sickle cells. Adults with sickle cell disease often experience reduce kidney function which can progress to kidney failure .

Patients with chronic renal failure often present to the ED with an unrelated condition. In these cases , the level of renal function may have important implications for diagnosis and treatment .
Valukas also cited " colorable " claims against Ernst &Young " for among other things its failure to question and challenge improper or inadequate disclosure in those financial statements. "
An unsuccessful candidate for promotion to an APS1-6 job can seek review of the decision to the Merit Protection Commissioner for failure to properly apply the merit principle. See jobs .
Pulmonary edema may be caused by damage directly to the lung , such as that caused by poisonous gas or severe infection. Lung damage with a buildup of body fluid is also seen in kidney failure .
14 : 30 : 55 Helicopter over jungle. Firing rockets from helicopter. Yellow signal smoke on ground. Troops crouch on edge of field .
fc : " I continue to wait for someone in the denier community , evolution , atmospheric science , or any other scientific field to come forward with a comprehensive theory that matches the data. "
JEREMY BLOXHAM : There ’s really no question about whether the Earth ’s magnetic field will reverse again. The question is not , if that ’s going to happen , it ’s when that ’s going to happen .
i. All corps artillery brigade , ACR , separate brigade , battalion and company CPs that may maneuver within the CZ (aviation , ADA , engineer , signal and MI) deploy TOCs to field locations .
base camp : a resupply base for field units and a location for headquarters of brigade or division size units , artillery batteries and air fields. Also known as the rear area .
Defines a CSS class name for individual fields that get disabled by this API. Note that a declaration at the HTML element level prevails to the values set via this function .
... mind could n’t find the write words to speak. The worse part is the CDC and medical field ca n’t figure out the right care for this disease. So , everyone suffers from that. < /p >
Flight noise is an issue even at subsonic velocities. The slow speed and lack of ballistic crack actually make it very easy to track the flight path of a subsonic bullet back to the shooter .

Most all RC airplanes I have flown will also display this. Most seem to slightly pitch the nose down (one would do so violently , even in level flight , not a good thing close to ground !) .
As I have a function to attend that night I ponder my options. Pay up and hope for the best , wait for the flight or ring Jetstar. I ring Jetstar AUSTRALIA .
Flight Performance Model - Developed from scratch by our own aeronautics professional , the flight characteristics operate within a narrow margin of operator ’s specifications
102.As a former chief flight instructor with 5 , 191 hours in the 767 , the relief first officer should have been readily able to regain control of the airplane .
Article 80 The vertical separations for airway , air route or ferry flight are based on flight level allocation. Flight levels are determined by the following criteria :
We would mark the ’Satisfactory ’ box for the Task 3.4 ’Communicate with flight crew’. The first officer and the captain effectively communicated this information .
Build a solid foundation of accounting concepts , skills and practical applications to prepare yourself for a wide array of opportunities in the field of accounting. Click here to learn more .
EACHBOX and BOX The list function i. applies to an argument such as 3 4 to produce a table , but we may also wish to apply it to each of the arguments 3 and 4 to produce separate lists. Thus :
Author ’s note : the reference_wrapper class template and the ref and cref function templates are shared with the Tuples library proposal. The text is repeated in both proposals .
Hohl D , Mehrel T , Lichti U et al. Characterization of human loricrin. Structure and function of a new class of epidermal cell envelope proteins. J Biol Chem 1991; 266 : 6626-36 .
(1) To interrupt or leave a computer program by pressing a specific key or a sequence of keys. (2) An option or function key displayed on a screen that allows you to access another screen .
I do n’t konw how to explain my idea about " generation gap " .i have n’t thought this question really before , but now i have to think it as my answer to my homework , this way is flesh .

Larter , S. , and R. di Primio. 2005. Effects of biodegradation on oil and gas field PVT properties and the origin of oil rimmed gas accumulations. Org. Geochem. 36 : 299-310. [CrossRef] 26
Hydroplaning - Take your foot off of the gas pedal , and keep the steering wheel straight. When the car 's momentum has slowed down and the tires grip the road again , you can regain control .
Will I have one but I ' m still not sure if its true or not.Thats is why I plan on sharing this story with you guys to see if you can help me figure out if what happen in my old house true or not .
I don ' t think that scenario is going to happen naman wala nang communist threat eh. People and the international community won ' t take it naman hanggang 2010 lang .
Remodeled 2 bdrm , 2 bath house , 2500 sq. ft. , w/ large spa tub in lower level bathroom , nice front & back yard , Internet access , &cable television .
Four in ten house sales are collapsing in some areas as buyers back out or fail to get a mortgage , the Bank of England has disclosed ... ..And this in an already slow market ... ..
I saw such built-in arena flashes at a USC basketball game years back , but have no idea how that one photog got access to them. Maybe he was the house photographer .
The only place I might disagree is the idea of the 0 or Void being a pole. The Void I think underlies ALL things – - from the movement of atoms to the shape of galaxies .
The challenge is to use public procurement and public services to lead the way , shape the market for innovative solutions and equip society to meet the challenges of the future. Innovation Nation
" Grandmother , grandmother , the stick will not break. The huge , huge duck wo n't lay an egg. She lifts her rear into the air and she wo n't let me swat her. What is an old , old man to do ? "
You do not have to apply for negative margin protection. Negative margins are a separate calculation and are automatically calculated when you apply for AgriStability .
Web browsers add extra space around a table to offset it from the top and left side of the browser window. To display your table flush against the edge of the window , set the margin properties .

FIG. 5 shows the screen of FIG. 2 with the pointing cursor at the right edge of the diagram space and the value of the right margin parameter indicating the desired measurement;
First , we set the length extrawidth to be the width of the gap between the main body of text and the margin paragraph area , and add to that the width of the margin paragraph area itself .
Relative strength of the DSL signal to Noise ratio. The higher the number the better for this measurement. In some instances interleaving can help raise the noise margin to an acceptable level .
In your business you have a figure called your operating profit , or net profit. If you divide this into your total revenue you obtain your operating profit percentage or net profit margin .
The margin of tolerance between an error-free picture and total failure (black screen) at all is very small. In such a scenario the variation until the picture is lost is little more than 1-2dB .
Do note that only securities that come under SEBI ’ s ’ Group 1 ’ are eligible for margin funding. These would generally be scrips that display low volatility and high liquidity .
This flag is set in the nsBlockReflowState constructor when reflowing a block margin root” frame (i.e. , a frame with the NS_BLOCK_MARGIN_ROOT flag set , for which margins apply by default) .
As you know , each note has a corresponding reference mark in the document main text. When you place the mouse pointer over a reference mark , the mouse pointer changes shape and looks like this :
Where you see the Quality Mark logo you can feel confident you will receive services of a high standard. The organisation awarded the quality mark will also display what level they operate at .
Scope of 10. (a) A trade mark must be registered in respect of particular application of goods or classes of goods. registration. (b) Any question as to the class within which any good fall
3. The patent body can under the own initiative or at the request of the owner to introduce to registration of a trade mark corrections of grammatical , typographical and other obvious mistakes .
pp. 52-53 : Underlinings , question mark in margin next to " French Declaration of the Rights of Man , " " obscure in margin next to " recondite. "

p. 49 : " 100 , 000 soldiers " in top margin. " Bile " is underlined , question mark in margin. " Wroth " is underlined. " Cap-a-pie " is underlined , check mark in margin .
California real estate market experiences a cathartic event that will lay the foundation for a recovery , Moody ’ s Economy.com In Investing on July 31 , 2008 at 8 : 17 pm
I for one ca n’t wait to see what the next outrageous financial market correction will look like. I know this much - this capitalist system , the future of glorious western society , is a failure .
Displays Help Launch Green Point Wines Moet &Chandon was looking for a dramatic two-case display to introduce Green Point , a premium category wine into the highly competitive U.S. market .
1.1.11 The ten global themes within the Charter suggest how Birmingham can pursue its ambitions to operate and compete successfully within the global competitive market .
Now it ’s even easier to manage your money no matter where you are. All you need is a web-enabled cell phone or PDA to access your checking , savings , CDs , money market , or loan accounts .
A vehicle ’s Black Book number is used to establish its market price and trade-in value. It helps to level the playing field for consumers , while still ensuring dealers receive a fair price .
Until we bid it a firm goodbye , this giddy market is an accident waiting to happen , and the giddier it gets and the longer the wait , the worse the accident. (Alan Abelson , 10/6/3) You can flood your mind and body with blissful , natural highs at any moment in time with this unknown method ... You can override all other emotions and override any amount of pain or fear !
" I acknowledge the privilege of being alive in a human body at this moment , endowed with senses , memories , emotions , thoughts , and the space of mind in its wisdom aspect. "
What annoyed me was the subtitle. With a subtitle like ’How the mind forgets and remembers ’ , I ’m afraid I have to conclude that this book does an injustice to the vast field of memory research .
Please keep in mind that , regardless of your expiration time , you can only park in a 15- or 30-minute parking space for the stated time limit. All other parking regulations still apply .

We only use 10 of the power of our subconscious mind power. Only 10-percent ! Can you imagine what would happen if your could access the remaining 90 ? The results would be incredible !
If you are deadly serious about your training and want to achieve your ultimate potential then our subliminal CD is for you ! It will instill in you the mind frame shared by all top level athletes .
To perceive with a special effort of the senses or the mind : descry , detect , discern , distinguish , mark , note , notice , observe , remark , see. See knowledge. see .
To do is the body 's wisdom. To think about what to do next is the mind 's wisdom. To know what to do at every moment is mind 's wisdom in spirit. To be is the wisdom of the pure spirit .
Donovan chose this moment to walk around his desk and park himself on one of its corners. Leaning over , he said , " There 's a lot at stake here , Miss Pappas. I wo n't lie to you about that. "
Q : Paul Stoddart has said he wants to accommodate a taller driver next year , but he has n't said what engine you 're going to have. There must be one or two things up in the air at the moment .
I wanted my blog readers to know that our dear friends are safe in another state while these crazy fires burn on. Their house is OK for the moment as well , but who knows what will happen ?
If we can experience every precious present moment then we will never have to endure constant suffering. Suffering starts from blocked energies manifesting in the mind , body and heart .
guided tours to mount cook national park including tasman glacier tours and guided wilderness hikes or walks. New Alps2ride fantastic scenic helicopter flight and easy mountain bike ride. L ...
We are seeking to either share a space with another movement organization or rent an 800-1 , 000 square ft. office (at below market rates) , preferably in Lower Manhattan .
Greek radio and television stations reported that the air force pilots saw no movement in the cockpit of the plane before the crash. There were some reports the two pilots seemed to be unconscious .
The ultimate advanced aerobics dance class requiring a seasoned mind , body and soul. Movement vocabulary consisting of , but never limited to : jazz dance , traditional aerobics and theatre .

A mid-16th-century movement , Italian in origin , although El Greco was a major practitioner of the style. The human figure , distorted and elongated , was the most frequent subject .
7. How does the MCGMAP address major challenges related to goods movement such as traffic congestion , environmental impacts , community concerns , and economic vitality for the region ?
(2) NOTEBOOK DISPLAY : This is the full display format of an individual NOTE in a NOTEBOOK file. No lines of the note in question are excluded from the display .
Click on a park name in the map below to check its current visibility and air quality. Please note that JavaScript must be enabled in your browser in order to fully view the web camera pages .
Assuming no strong forgery threat model , such a note could be crafted with today 's PC printing capaibilities. Such a situation exists in small community currencies , those with O (1000) users .
Sticky note : Cover with card paper Size : 12*15.5CM Weight : 70g for 60pages Packing : OPP or shrink. Using : Convenience to write , stick repeat , Perfect for promotion , advertisement , gift. ...
This is a security level assigned to each note which controls who can and can not access this note. This value is established in the employee set-up. Refer to [Link] for further information .
The PREVIOUS command is used in the NOTEBOOK environment to display the previous note if full. It is used from NOTEBOOK DISPLAY as a " previous note " function .
" lowest note of a musical scale , basis of a scale , " 1776 , from key (1) in sense of " musical scale " + note. Fig. sense of " leading idea " is from 1783; keynote address is 1905 , Amer.Eng .
I am a level one dental assistant with over 5 years experience in the field seeking full-time or part-time work in an office within the whitby/oshawa area. I have experience working with both a ...
A major project at the moment is the three-story , 200 , 000-square-foot office building being completed off Stelzer Rd. in the southern edge of Easton to replace the Gahanna location .
Hi my name is Leni , 31. Looking for a full time work as office assistant/secretary position , entry level position I have years of experience in Administrative works e.g. clerical &general ...

Law enforcement is a very dangerous job. At any given moment , a situation can arise where an officer ’s life is in danger and he needs to defend himself. While a good officer needs to rely on ...
You did address it (sort of) , but left it hanging on a question. It does give us an idea of what have been going through the officers mind , but that still does n’t address policy .
Burns is a graduate of Villanova University who received his commission through NROTC in 1988. He entered the aviation training pipeline and was designated a naval flight officer in 1989 .
A ’Pay and Display ’ system of parking charges apply at Daventry Country Park. This applies to the main car park and the overflow parking field (both accessed from Northern Way) .
The split level car park is a highly efficient shape , especially suited to a site with a slope of 1.5m over the 32m width. The longer the better for efficiency , it will normally be 32m wide .
They worry about increased traffic , noise and trash the site will generate. They have concerns about the ball field ’ s impact on the park ’ s wildlife. And they question the cost of the field .
The journey through the short tunnel to our right field seats was almost surreal. Hustling through that corridor seeing only the blue sky , I couldn ’ t wait to get a full view of the park .
Please note that there is a height restriction of 1.95m in this Car Park. Any vehicle in excess of this height will be required to park in the Long Stay Car Park where no such restrictions apply .
To let - modern first floor office premises on prime business park north of Carlisle. Easy access to motorway network. On-site car parking - 10 spaces. 2 , 263 sq ft (210 sq m)
Subdivision 2 [Baseball Park.] assigns the strategic and long-range planning office to assist a baseball site commission in a process to provide for the construction of an open-air baseball park .
Long said , " Almost all the money coming into the Vietnamese stock market is coming from overseas funds. It is really a good signal for the market when US investors want to pour more money in. "
Polio said Buker , who has since died , told him the district attorney ’s office had decided not to pursue the charges. Polio had no reason to question it at the time , he said yesterday .

Principals have the authority to control the day-to-day running of a school. This authority probably gives the principal the right to take you out of class and question you in their office .
we will then take to the streets to the esplanade where we will assemble for an aerial photo in 350.org formation ! then we cruise to parliament house , to meet the other 350.org rally .
Every cell in your body is alive , burning with excitement , the adrenalin pumping through your veins. Only few experience the unbelievable adrenalin that is rally driving .
One of the largest campaign contributors to U.S. Congressman Mark Kirk would reap huge financial benefits after passage of the Waxman-Markey climate control legislation (" cap &trade ") .
In this guided relaxation download process you will consciously relax your mind and body as you prepare yourself to go into a deep , peaceful , rest-filled sleep. 25 minutes
Now minimally tighten your left fist. Hold at this level so that you just feel the tightening. Let go and relax the fist. Let the relaxation spread through the arms and the rest of the body .
Renown , n. Distinction , repute , name , reputation , note , notoriety , notability , figure , fame , celebrity , glory , honor , eminence , lustre .
Suggest that if you wish local people to accept the road then you provide them with access to it. Reducing speed limits and improving slip roads would allow some of the side roads to be kept open .
Rooting hormones can be used to improve or speed up rooting in some plants. Various commercial formulations are available. Most house plants root readily and rooting hormones are seldom necessary .
1.Solve project is RK chip 2.2.4 inch TFT screen display 220*176 image 3.MPEG4 AVI formart playing 24 frame/s 4.Suport Mini SD card 256M 2GB
Tempered safety glass provides the highest possible level of weather resistance while adding a touch of class to your door. Glass can be added to any door , regardless of your screen selection .
A slanting surface or bevel on the edge surround of a lens. Also a groove or flange designed to hold a lable or information plate that may surround a lens. The frame around a CRT or LCD screen .
Amazon.com : nEW 4GB 3 " QVGA screen display (16 : 9) Touch Screen MP5 MP4 Mp3 GAME RMVB RM Player + support mini SD card to extend+ MIC and FM recording function : Electronics

Darth Maul Screen Saver v1.0 - Lets you display the haunting image of Star Wars ’ Darth Maul as a floating screen saver. Please note that the screen saver also runs in Windows 3.1. (12/1) download
Select the Delete button to delete the displayed access device entry. You can select the Cancel button and return to the Access Device screen without deleting the entry .
Referring to FIG. 1 , the touch screen display device 1 includes a touch screen display panel 300 , a signal controller 400 , a gate driver 600 , a data driver 500 , and a read-out unit 700 .
5. The computing device of claim 4 wherein the second display panel is a flexible substrate screen and is configured to be rolled into a tube shape upon deployment in the folded-in configuration .
Simple screen saver. Mathematic transformations of the flying figure with color flowing. Several options to change the visual effects (phantom , speed etc.) . Preview and test mode support .
Lean the entire display assembly (LCD screen and rear plastic bezel) upright 90 degrees. Now gently lean just the LCD screen forward to lay on the paper towel or cloth covering the keyboard .
I think I know that I ’m female because I ’ve always been told that I am. Because my community sees me as female , and because the shape of my body says to my community : this is a woman-shape .
The shape element is a generic element. The characteristics of the shape is defined in a separate Java class. The shape can be anything from a simple rectangle to a complicated figure .
3. human body , physical body , material body , soma , build , figure , physique , anatomy , shape , bod , chassis , frame , form , flesh , body , organic structure , physical structure
When you remove your leg from the water to shave , the exposure to the air can cause goosebumps. Unless you want a red post-shave rash , wait to shave after the goosebumps have disappeared .
Less commonly , speed limits are reduced to address local air quality issues or other factors affecting environmental quality. An example is " environmental speed limits " in the United States .
David , C. T. (1978) . The relationship between body angle and flight speed in free-flying Drosophila. Physiol. Entomol. 3 , 191 -195. [CrossRef]

If strength exercises are used to help develop speed then the weight must be low enough to allow the same speed of action as the play displays. Otherwise , the movements slow down on the field .

7. All road signs indicating speed limits or weight limits shall be uniform in size , shape , lettering and coloring and shall conform to standards established by the department of transportation .

But the DCT-700 still has to decode the signal and figure out what it is. I have one of these in my office , and it really is n’t much different in speed than the HD-DVR or the standard SD box .

(2) In level flight at any speed from VH or VMO/MMO , whichever is lower , to either Vx or 1.4 Vs1 , with the landing gear and wing flaps retracted .

A class for children who are swimming less than 10 feet independently. Introduces floating and beginner swim skills in a relaxed environment at the child ’s own emotional and developmental level .

Its huge 50 inch screen means that you can put it in the corner of a large front room for everyone to enjoy. Movies , games and television will never be the same ever again in your house !

The term Rabbit ears is used to describe an antenna that sits on top of the television set to receive over-the-air signals. The antenna arms stick out in a V shape , much like a rabbit ’s ears do .

In some markets , you will need to have a DIRECTV Multi-Satellite System. Can DIRECTV be connected to more than one television set in my house ? Buying separate DIRECTV Receivers for ...

LG Electronics India Ltd (LGEIL) plans to introduce in the market the country ’s first direct-to-home television sets by February or March. Read More ...

White Dot : White Dot is the international campaign against television , based in Chicago and Brighton. We seek to answer that ridiculous question : " What do you do if you do n’t watch TV ? "

The International Criminal Court should deny veto rights to the US government. The threat of class actions against US government departments may then lead to the end of these atrocities .

I weigh 95 pounds. I haven ’ t been under 100 pounds since elementary school because of my boobs. Woahhhh. It ’ s kind of an exciting , crazy idea for me to comprehend at the moment .

To weigh an item , simply attach the scales to the item using the strap and lift. The automatic hold function will allow you to set the item down and it 's weight will still be displayed on screen .
Lt. Gen. Fuh Wei-ku , commandant of the Air Command and General Staff College at the Armed Forces University , notes that for the moment Taiwan still has a slight edge in terms of air defense .
In response to your overly optimistic , outdated piece on how long unemployment lasts -LRB- People Patterns , Sept. 20 -RRB- : I am in the communications field , above entry level .
Developing the company -LRB- and extremely fast at that -RRB- is a road that must be taken , and market share is an indication of the speed of the company 's development .

Table B.1: complete dataset used in performance evaluation