INSTITUTO POLITÉCNICO NACIONAL

Centro de Investigación en Computación

Laboratorio de Lenguaje Natural y Procesamiento de Texto

"KNOWLEDGE-RICH TECHNIQUES FOR ANAPHORA RESOLUTION"

TESIS

Que para obtener el grado de:
Maestro en Ciencias de la Computación

Presenta:

Jesus Alexander Alvarado Gutiérrez

Directores de tesis:

Dr. Alexander Gelbukh

Dra. Olga Kolesnikova

México, Ciudad de México
Diciembre, 2016

A mis padres Lilia y Jesús
quienes siempre me han brindado
su amor y confianza

A mi fuente de inspiración
quien con su amor
me motiva a siempre dar lo mejor, Brenda

Acknowledgement

Gracias al Dr. Alexander y a la Dr. Olga, quienes sin dudarlo me compartieron de su conocimiento y apoyo para realizar esta tesis

Al Dr. Hiram, por iniciarme en el área del procesamiento del lenguaje natural y a quien le debo las primeras enseñanzas en el Centro de Investigación en Computación.

A mis padres Lilia y Jesus, quienes siempre han estado a mi lado, que con su amor y cariño me dan la fortaleza necesaria para seguir en este duro camino de la vida y superar cada reto que se presenta.

Gracias a mis amigos del Centro de Investigación en Computación, quienes, a su manera, me aconsejaban para trabajar más duro.

Al ahora ya maestro en ciencias de la computación José Aguilar, ya que, sin su aporte, enseñanzas y amistad esta tesis seguiría escrita en Word.

Al Instituto Politécnico Nacional por acogerme como alumno y al CONACYT por apoyo económico para facilitar el término de esta tesis.

Por último, pero no menos importante a Brenda, mi inspiración y mi fuerza para superarme día a día, por brindarme tiempo, comprensión y amor, pero sobre todo por darle luz a mi vida.

Resumen

Esta tesis se basa en el problema de solución de anáfora, el fenómeno de la anáfora se presenta cuando se quiere economizar en palabras y se hace la omisión de un elemento ya nombrado con anterioridad en la oración, y en su lugar se coloca un referente. Esta tesis se basa en la relación anafórica presente entre sustantivo y pronombre en el esquema de Winograd. La resolución del antecedente para el pronombre puede ser usada en diferentes tareas de procesamiento de lenguaje natural, como generación de resúmenes y la traducción automática.

Existen diferentes enfoques para abordar el problema de resolución de anáfora y mas específicamente en la resolución de pronombres, estos enfoques están basados en conocimiento,o en inferencia lógica. El trabajo realizado en esta tesis se encuentra en el primer enfoque, el enfoque basado en conocimiento.

Esta tesis propone diferentes métodos para tratar de encontrar el antecedente correcto para el pronombre en la relación anafórica, usando técnicas de eliminación de sufijos y prefijos en las palabras, así como los determinantes en los antecedentes, incluyendo sinónimos e hiperónimos para así tener un mayor rango de búsqueda en la relación verbo-sustanvo.

Por ultimo, el trabajo en esta tesis fue evaluado con el corpus del evento “Winograd Scheme Challenge” que es organizado por Naunce Communication, este corpus es frecuentemente usado por las sistemas similares y muestra que es posible obtener un mejor desempeño para algunos en el estado del arte

Abstract

This thesis is based the anaphora resolution problem, the anaphora is the presupposition of an element so this element is not mentioned directly, so the anaphora is a reference to previous antecedent. so the anaphora resolution problem try to find the antecedent of a referent, this thesis is focused in the anaphoric relation between a noun and a pronoun in the Winograd Scheme, the anaphora resolution can help to many task in Natural Language Processing, such automatic translation, automatic summarization, etc.

Nowdays there are many different approaches to tray to solve the anaphora resolution and more specific in pronoun resolution, there are knowledge and logical inference approaches with supervised or unsupervised methods. This work is based in a knowledge approach with similarity and relatedness measure.

This work proposed many methods to try to choose the correct antecedent for the pronoun in anaphoric relation, using stemming,removing determinants, synonyms and hypernyms to get a higher range of match in verb-noun relation.

Finally this work is evaluate with the corpus of Winograd Scheme Challenge, that is an event that is organized by Nuance Communication, this corpus is frequently used to train and test similar systems, showing that is possible to get a higher than others in the state of art

Contents

List of Figures

List of Algorithms

List of Tables

Chapter 1

Introduction

Natural language is the way that the humans use to communicate either oral or written way, this language is spontaneous generated without stablished rules and it is continuously involving to meet the social or cultural needs of communication between society, as well as the ways to communicate between humans and computers.

Whereby computational linguistic is the interdisciplinary science that wants to give capacities of to understand, to use and to generate natural language to these devises, making use of the task of Natural Language Processing (NLP) because the main objective is to create models for the natural language and thus to be able to use them in the computers.

Nowadays, great results have been achieved and there have been a great progress in many problems that are part of study of the NLP, but many else still need more work, for example anaphora resolution, coreference resolution and word sense disambiguation are the three most important problems in the PNL, the first is anaphora resolution this task aims to identify the interpretation of an element that depends of an antecedent that was previously mentioned in a statement, the anaphora relationship is generally present in a noun phrase between nouns and a pronouns, and this pronoun make reference to an entity in the real life. the second one is coreference resolution task that aim to identify all nominal phrases that make reference to the same element in the real life. and the last is word sense disambiguation task that aim to identify what of all meanings or sense of a word is correct inside a certain context in the statement.

The proposed method in this thesis is focussed in anaphora resolution problem seeing this like a specific problem of a word sense disambiguation problem.

1.1 Problem statement

One of the mean task in the Natural language Processing is the resolution of pronoun that is define like the anaphoric relation that exists between a profound and a noun that have been previously mentioned (antecedent) in a noun phrase. The importance of this task is because there are many anaphoric relations in the text and its correct resolution is necessary in others task like automatic summarization, automatic translation and in Paraphrase Identification.

This thesis take a specific pronoun problem that is known like a pronoun resolution this task have a challenge called "Winograd schema" because the main task in this problem is resolve the pronoun in each schema, for example.

Definition 1. *Winograd schema*

1. S_1 : *The bee landed on the flower because it had pollen.*

2. *P_1 : it*
3. *C_1 : the bee*
4. *C_2 : the flower*
5. *A_1 : the flower*

where

1. *S_1 : Complete statement (with antecedents and pronoun)*
2. *P_1 : Pronound to be disambiguate and is present in the statement S_1*
3. *C_1 : First antecedent that is mentioned in the statement S_1*
4. *C_2 : Second antecedent that is mentioned in the statement S_2*
5. *A_1 : correct antecedent for the pronoun P_1*

1.2 Objectives

1.2.1 General objective

The general objective of this thesis is to develop an algorithm to find the correct antecedent of a pronoun that appears in an anaphoric relation. Using knowledge techniques applied from WordNet and the Sketch Engine relation of verb and noun.

1.2.2 Particular objectives

1. To study and adapt the syntactic bigrams of constituents to get the elements en a statement

2. To implement complementaries techniques for preprocessing of the corpus.

3. To get the weight of relevance in a relation verb - noun to applied in antecedent resolution method

4. To get the semantic relations of synonyms and hypernyms of a noun to applied in antecedent resolution method

5. To use the different similarity and relatedness measure to evaluate the Winograd Scheme

6. To compare the results in an assemble algorithm

1.3 Significance of the study

The anaphora phenomenon is present in natural language that is used by humans. in the text is possible to omit an element, because it was already mentioned and do this serve to economize the used words, this problem is common when is using nouns that are referent to one element in the real word and when this element ids used again, generally is used a pronoun.

Identify this coreference between pronoun and noun is a hard problem, and so this problem is called Hard Coreference Pronoun, the resolution to this problem not have many sense, but is important in an others task in the Natural Language Processing (NLP) for example in automatic translate take an important relevance because in english the pronoun "it" is for both female and male, but in many languages this pronoun is different, so the correct antecedent identification give a correct type, in an other task where is util the pronoun resolution is in automatic summarized because in a text there are many coreference and anaphora phenomenon and its necessary the resolution for found the correct noun and so classified it.

The method proposed is for the english language because there are more semantic resources, computational dictionaries like WordNet that is more confidence in english. syntactic parser and corpus to do this task.

The method proposed have many modules than can serve in others task in the natural language processing, this modules are mentioned in the follow section.

1.3.1 Provied modules

Anaphora resolution is a process that need many resources and linguistic tools, such as corpus, computational resources, semantic labeled corpus, syntactic parsers, etc. and requires too many specific process such as similarity and relatedness measure, evaluation metrics and others, this work give many of this.

- Method to find the correct antecedent.
- Syntactic parser based in Stanford parser.
- Module to get the mixed noun and verb bigrams.
- Module to get the similar words like synonyms and hypernyms.
- File resource with verb-noun relation frequency.
- File resources with relevance of the relation verb-noun.

1.4 Document structure

Chapter 1 : Introduction. This chapter shows a brief introduction to the natural language and the natural language processing, beside is showed the problem statement in which is defined the problem and it restrictions, further in this chapter is showed the general objective and specific objectives, as well as the significance of study and the contributions.

Chapter 2 : Basic Background. This chapter describes the tasks of anaphora resolution, coreference and the differences between them, describes the Winograd Scheme Challenge, and describe basic information about the language, and the source and techniques used such as parsers and WordNet.

Chapter 3 : Methods for Pronoun resolution. This chapter describes the different works that have address the anaphora resolution, in specific to pronoun resolution and Winograd Scheme Challenge, this work are classified by task and techniques for resolution.

Chapter 4 : Proposed algorithm. This chapter describes the proposed methods, explain the similarity and relatedness measure as well as the process to have the weight of relation relevance, explain each module with her input and output and how interact with each others.

Chapter 5 : Experimental results. This chapter describes the obtained results and compare each result by measure and show the assemble result that show how the results are increased and a brief explanation of each measure.

Chapter 6 : Conclusions. Finally this chapter shows the conclusion of this thesis, contributions, and future work.

Chapter 2

Basic Background

2.1 Natural Language and Natural Language Processing

2.1.1 Natural language

It is not known exactly how the language arose, however the natural language used by humans has the general purpose of communication between them. Thanks to this communication was possible the evolution of the human species through the information, which can be transmitted orally, written, etc. In general, natural language conforms to the following levels (Sidorov G., 2001):

- *Phonetics / phonology:* phonetics is the branch of linguistics that studies sounds, the which is an essential factor of human language. Problems in computational phonetics are related to the development of systems of analysis and synthesis. Phonology describes the way sounds work, that is, you are interested in relationships with other sounds and their implications.
- *Morphology:* is the branch of linguistics that studies the internal structure of words such as suffixes, prefixes, roots and flexions, also studies the system of categories grammar of languages, such as gender, number, etc. The main problem of the morphology is that the implementation of morphological analysis and synthesis systems need to develop large root dictionaries, which is quite difficult to obtain.
- *Syntax:* part of the grammar that is responsible for studying the relationships between words of a sentence and the function that each performs in it, to represent such relationships there are two models: dependencies and constituents.
    - ◇ *Structure of constituents:* in this structure the relationships are presented in the form of a binary tree that shows how a sentence is divided into parts or constituents, it is important to mention that each language has an established order for constituents (basic units of the sentence).
    - ◇ *Dependency structure:* here relations are marked with arrows on the tree of parsing dependencies, a word may have several depend on it, and however, each word may have only one dominant. In this structure it is shown how words are linked together within a sentence.

    Computational syntax must have methods for automatic analysis and synthesis, that is, construct the sentence structure, or generate the sentence based on its structure. The development of generators is a less complicated task and it is clear what algorithms are the needed for these systems. In contrast, the development of parsers (Also called syntactic parsers) remains an open problem, especially for languages that do not have a fixed word order, as is the case of Spanish. At English language, the order of words is fixed, so English-based theories are not so easily adaptable for Spanish.

- ◇ *Semantics:* is a subfield of linguistics whose purpose is to understand the meaning of a sentence, for which one must know the meaning of the words and give the interpretations to syntactic relations. Another task of semantics is to define the meanings of words, which is a complicated task because there are different semantic relationships among them, such as: hyponymy, synonymy, polysemy, homonymy and antonymy. The definitions of the meanings of words are found in dictionaries, however, the definitions of words are defined by other words, which can lead us to a vicious circle. As a result of the semantic analysis, networks semantics, which are a form of representation of linguistic knowledge, where the concepts and relations are shown by a graph that usually represents a hierarchy. The problems of computational semantics are very interesting, however many of these jobs still have no solution, so more research is needed do.
- ◇ *Pragmatics:* it is the branch of linguistics that is interested in the context that influences the interpretation of the meaning of a sentence. A sentence may have intentions or different interpretations in different contexts. As we already have many problems in the semantics, we can not continue the analysis at the next level, so we must take this into account.

2.1.2 Natural Language Processing

Also called computational linguistics, Natural Language Processing is a discipline as old as the use of computers, the latter can be very useful when you want to process a huge amount of data, however, you are still not able to understand all the information provided by the data, so the use of technologies of the PLN are made necessary. Some of its applications are the following:

- Information Retrieval: The goal of this application is to provide the user relevant information about a specific request, this work is not easy because of there may be a lot of information that far from being a positive thing can cause problems because it is difficult to find the relevant information.

- Interfaces in natural language: interfaces mean communication between the user and machines, where the user does not require any previous knowledge about the handling of programs, only natural language management is needed.

- Automatic translators usually translate the language intermediate representation (for example syntactic analysis trees), in which differences can be adjusted as the order of the words for later transform the intermediate representation into target language text. This application is not simple.

- Abstract: is the process of analyzing a large text or a collection of texts and generating a report containing the relevant information of the source texts in order to the reader has an idea of the content without the need to read all texts.

- Extraction of information: it is an application more that arose thanks to the appearance of great volumes of texts and their goal is to automatically extract structured information from readable documents to a computer.

2.2 Syntactic analysis

To the process of obtaining the syntactic relationships that words have within an expression written in natural language or, the way in which they are grouped, is known as syntactic analysis. The syntactic analysis can be studied in the form of grammars that define the structure of language, such as Context Free Grammars Grammars - CFG, Probabilistic Context Grammar Free Grammars - PCFG), Core-dominated Grammars (Head-Driven Phrase Structure Grammars - HPSG), etc ; On the other hand, the syntactic analysis can be carried out by means of computational structures such as the parse trees of which there are two different types: analysis trees of syntactic dependencies and parse trees of constituents. This work uses the parsers analysis of dependency and constituent trees, so that in the following subsections the main characteristics of these two types are described of analysis trees.

2.2.1 Syntactic analysis of constituents

This section describes constituent-based parsing, which is based on trees of syntactic analyzes, said trees describe a hierarchy formed by groups of words or syntactic constituents that form an expression written in natural language. Each group or constituent has associated a label that indicates the type of grouping, generally suggested by the group's main word.

As can be seen, in a tree of particle analysis of constituents the leaves of the tree are the words that form the original expression and the parents of the leaves correspond to the grammar category tag assigned to each of them, for example "Maria" and "Book" is labeled as a noun (NN), "reads" as verb (V), "the" as determinant (DT) and "infantile" as adjective (ADJ).

Intermediate nodes above those mentioned are assigned a group tag or a syntactic constituent within the sentence; Such is the case of the group of words "the book" that form the constituent Nominal Group (GN) or Noun Phrase (NP) in English, in turn the group of words "read the children's book" form the constituent Verbal Group (GV) or verb Phrase (VP) and so on until you find the constituents GN and GV that form the complete Sentence (O) in English. The syntax analysis trees of constituents are closely related to the grammars context Free Grammars (CFG) as they are used to represent a sequence of grammar rules applied successively to generate the expression in this sequence of rules is known as a derivation (Chomsky, 1956).

2.2.2 Syntactic analysis of dependencies

Dependency-based parsing has its origins in ancient Greece and in linguistics hindu by what is even older than the syntactic analysis of constituents (Jurafsky and Martin, 2009). Dependency analysis indicates the syntactic relationship between pairs of words associated within Of a sentence. More specifically this type of analysis indicates which word is dominant and which dependency (untyped dependencies) as well as the type of dependency relation that exists between them (typified dependencies). The trees in Figure 3.2 show the tree of analysis of dependencies not typified and typified for the expression "New cars use little gas". As it can be seen, in a dependency analysis tree, each node corresponds to a word of the sentence analyzed and each edge indicates a governing-dependent relationship, said relationship being indicated or not depending on whether said tree is typified by a label assigned to say edge.

It is not known exactly which of these types of analysis (constituents or dependencies) is the Better, however, the researchers in favor of the dependency analysis argue that this Type of analysis is favored in languages with "variable word order", which allows a word is placed in different positions of a sentence maintaining a correct syntax which hinders the task of analysis by constituents. Due to the above, they have developed procedures that make it possible to convert constituent trees into dependency trees (Xia and Palmer, 2001) algorithm) not typified by what can be observed a certain equivalence between the two representations.

2.2.3 Stanford Parser

The Stanford parser is a tool that enables parsing from expressions written in natural language. This analyzer has support for both types of formalism (dependencies and constituents), is currently widely used by natural Language Processing systems due to their good results and to their constant upgrade.

For the analysis of constituents, the analyzer is based on statistical methods from repositories Of trees (Klein and Manning, 2003), similar to approaches that use other analyzers (Collins, 1999) and Charniak (Charniak, 2000), and for the analysis of typed dependency parse. This analyzer is based on the representation of constituents and consists of 48 different types of dependency relationships, which are ordered in a hierarchy, these relations were initially based on the set of relationships defined in (Carroll et al., 1999) and (King et al., 2003), although over time they have been introduced refinements to them, since the primary focus of the analyzer is more practical than theoretical.

It is also important to mention that the Stanford parser generates output for the representation of dependencies, triplets composed by the grammatical relation that binds to a dependent word with its ruler according to the following form:

relación(gobernant-##, dependent-##)

Where ## refers to the number of the word within the input expression. Then The analyzer output is displayed for the expression "Mars has two known moons, Phobos And Deimos"

$nsubj(has-2, Mars-1)$
$root(ROOT-0, has-2)$
$num(moons-5, two-3)$
$amod(moons-5, known-4)$
$dobj(has-2, moons-5)$
$appos(moons-5, Phobos-7)$
$appos(moons-5, Deimos-9)$
$conj\_and(Phobos-7, Deimos-9)$

On the other hand, the representation of the syntax analysis trees of constituents follows a notation of nested lists.

2.3 N-grams and syntactic N-grams

In this section we present the traditional N-grams and the syntactic N-grams, highlighting, Its main characteristics, uses and classification.

2.3.1 N-grams

A strict definition of N-grams is as follows: an N-gram is a sequence of n words of an expression in natural language, and are generally used in processing applications such as automatic correction, voice recognition, and automatic translation, grammatical category labeling, natural language generation, similarity between words, identification of authorship among others (Jurafsky and Martin, 2009). An example of possible sequences of 2 words or bigramas (2-grams) for the expression "Friday I have exam" Are as follows: **On Friday Friday I have I have an exam** And the possible sequence of 3 words or trigrams (3-grams) are: **On Friday I have Friday I have an exam.**

2.3.2 Syntactic N-grams

concept of syntactic N-grams (syntactic N-grams / sn-grams) which are N-grams or word sequences (or other elements) obtained from the order in which the elements appear in the syntax analysis trees (Dependencies or constituents) of a sentence; More specifically, the syntactic N-grams are constructed from the sequence of nodes that can be reached in any path of length n in a parse tree. This type of syntactic N-grams is known as continuous syntactic N-grams (Sidorov et al., 2012).

There are different types of syntactic N-grams classified according to the type of elements that These take into account. The following is mentioned:

- **Word sn-grams:** in this type of syntactic N-grams elements that are taken in count are the words that constitute the input expression.
- **POS sn-grams:** In this type of syntactic N-grams, the labels of grammar category corresponding to each word of the input expression in place to take those words directly into account.
- **Syntantic relations sn-grams:** In this type of syntactic N-grams, the elements that they are taken into account are the labels assigned to edges that join two words within the dependency parsing tree. These labels correspond to the syntactic relations between the words that this edge unites.
- **Mixed sn-grams:** this type of syntactic N-grams takes into account a mixture of different types of elements such as: words, grammatical category tags and / or types of syntactic relationships.

2.4 Stemming

Stemming consists of removing non-essential parts in words such as suffixes or prefixes in such a way that the remaining fraction of a word after this process is known as "stem" or lexeme. It should be mentioned that in English there are the concepts of lemmatization and stemming. In the process of stemming a set of rules is applied that allow to realize this reduction, so this process is responsible for obtaining the stem of a word which is not necessarily has a meaning, an example of the stemming process is as follows:

- Dog, puppy, dogs → dog

On the other hand, in the process of lemmatization a morphological analysis of the word with order to bring it to its base form (lemma in English) (Manning et al., 2009) in such a way that said base form has meaning. An example of this is the following:

- I ate, I ate, I ate → Eat

Stemming and stemming techniques are useful in areas such as information retrieval, where it is desired to retrieve data that matches a given word, helping Increase the retrieval of important documents. A disadvantage is the reduction in the efficiency of the systems, which is affected due to the resources required to implement these processes. There are several stemming algorithms; However one of the most popular is that of Porter (Porter, 1980), who uses a set of rules with the purpose of eliminating certain suffixes of words; Algorithm is popular for its speed and good results.

2.5 Stop words

Another technique that is very helpful in some applications of Language Processing natural is the elimination of stop words. These words refer to words that occur very often in expressions of a language, so the appearance of them does not information, and therefore, are often eliminated before or after processing a natural language sentence to be analyzed.

There is no definitive list of stop words since it depends on the type of application, however the most common include prepositions and articles as part of these. Then provides a list of the most common stop words for the English language:

a, able, about, across, after, all, almost, also, am, among, an, and, any, are, as, at, be, because,been, but, by, can, cannot, could, dear, did, do, does, either, else, ever, every, for, from, get, got,had, has, have, he, her, hers, him, his, how, however, i, if, in, into, is, it, its, just, least, let, like,likely, may, me, might, most, must, my, neither, no, nor, not, of, off, often, on, only, or, other,our, own, rather, said, say, says, she, should, since, so, some, than, that, the, their, them, then,there, these, they, this, tis, to, too, twas, us, wants, was, we, were, what, when, where, which,while, who, whom, why, will, with, would, yet, you, your.

2.6 Lexical semantic relations

Semantic linguistics is a subfield of semantics in general that studies the meaning of The linguistic expressions that do not depend on the context. on the other hand, lexical semantics deals with the study of relationships between words, which are seen as nodes within a graph or hierarchy, from which it is possible to obtain semantic lexical relations such as the following:

- *yperonymy:* hyperonymy is a lexical semantic relationship that can be thought as a generalization relation. For example, tree is hyperonym of cedar, since the latter includes all tree features.

- *Hyponym:* unlike hyperonymy, hyponymy can be seen as a relationship of specification, so that Mondays and Tuesdays are hyponyms of day, since these contain in their meaning day features.

- *Synonymy:* refers to the relationship between words of similar meanings but which are Written in a different way and therefore are interchangeable without altering the meaning Of a sentence. An example of this are the words auto and car.

- *Antonyms*: it is the opposite relation to the synonymy, in which the meaning of two words Is opposite. An example of this is: black and white.

- *Meronys*: can be seen as a relationship "is part of". For example the word finger It is hand merimo.

- *Holonyms*: is a relationship similar to meronimia that indicates the relationship "composed of". an example of this is the word tree that is holonomic of leaf, branches and trunk.

2.7 Similarity between words

Lexical semantic relations such as the case of synonymy (described in the previous section), can be seen as a binary relationship, since two words may or may not be synonymous; so, these relationships can be seen as rigid concepts. Due to the above, many applications use a more flexible metric called word-likeness, which based on the intuition that two concepts are more similar the greater the number of characteristics that they have in common, and conversely, two concepts are semantically different or have a greater semantic distance, as their common characteristics are less.

Word resemblance calculation is useful in many applications, such as: retrieval of information, question answer systems, automatic summaries, language generation natural, automatic translation, etc. The methods that try to measure the similarity between words are divided into two: based on thesauri and distributional methods; The first measure the distance between two words using the graph formed by the hierarchy of words of the thesauri, while the second esteem the similarity between words giving like similar words that have distributions similar in a corpus (Jurafsky and Martin, 2009). Some measures are described below. Based on thesaurus since they are the ones used in this work of thesis.

2.7.1 Basic Concepts

In order to describe thesaurus-based similarity measures it is important to define three Concepts:

- **Probability of a concept:** this is calculated by counting the number of times the concept "c" in a corpus and adds to the number of times you see any other concept derived from "c" therein. Finally this amount is divided by the total number of words in the corpus as shown in the following formula. It should be mentioned that estimation of this value requires an additional corpus to the thesaurus (Resnik, 1995).

$$P(c) = \frac{\Sigma_{w_c words(c)} count(c)}{N} \tag{2.1}$$

- **Information Content (IC):** refers to the amount of information contained in a concept and is calculated by obtaining the negative logarithm of the probability of this concept within the thesaurus hierarchy

$$IC(c) = -logP(c) \tag{2.2}$$

- **Lowest Common Subsumer (LCS):** Taking into account the Hierarchy of a thesaurus, the closest common ancestor is the lowest node in the hierarchy Which is achieved by two given concepts.

2.7.2 Path-length

The measure of path length similarity (Leacock et al., 1998) is based entirely on the intuition that two concepts belonging to the hierarchy of a thesaurus are more similar while their

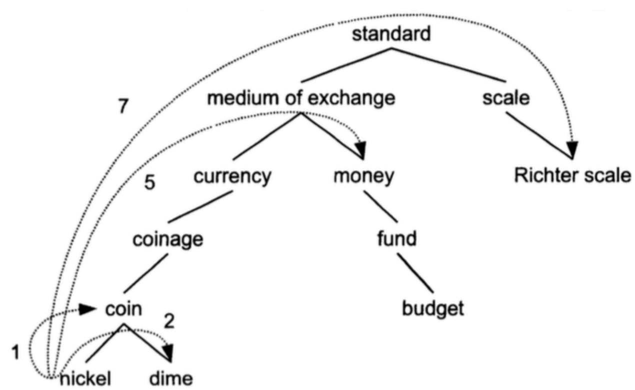

Figure 2.1: path length

distance within the hierarchy is smaller. To measure the distance between two concepts is counted the number of edges by which these concepts are separated taking into account the shortest possible path between the two nodes belonging to them. The figure 2.1 graphically shows the values corresponding to the distance between the concept nickel and the other words within a mini hierarchy. Taken from (Jurafsky and Martin, 2009).

2.7.3 Resnik Similarity Measure

The measure of similarity length of travel, has the disadvantage of taking uniformly the distance between concepts so that, according to this, the distance from nickel to money would be the same as from nickel to standard, taking as reference to Figure 3.11, which is not very intuitive. Due to the above, other measures were created that try to solve this problem based on corpus that allow the calculation of information content values for each concept. The first of these is Resnik's resemblance measure (Resnik, 1995), which considers that the similarity between two concepts is equal to the information content of the ancestor as shown in the following formula (2.3):

$$sim_{resnik}(c_1, c_2) = IC(LCS(c_1, c_2)) = -log(LCS(c_1, c_2)) \tag{2.3}$$

2.7.4 Lin Similarity Measure

This measure (Lin, 1998a) is based on the intuition that to measure the similarity between two concepts A and B, it is necessary not only to measure the common information between them, but also the difference between them. According to the above, the similarity according to Lin is obtained measuring what two concepts A and B have in common as $IC(common(A, B))$ and the difference between them as $IC(Description(A, B)) - IC(Common(A, B))$. The Similarity Formula proposed by Lin is shown below (2.4):

$$sim_{lin}(c_1, c_2) = \frac{2 \times logP(LCS(c_1, c_2))}{logP(c_1) + logP(c_2)} \tag{2.4}$$

:

2.7.5 Jiang-Conrath Similarity Measure

The Jiang-Conrath measure (Jiang and Conrath, 1997) propose a measure of similar similarity To Lin's, although derived in a different way, since it expresses the similarity function taking into account the distance between the concepts, as shown in the following formula (2.5):

$$dist_{jcn}(c_1, c_2) = 2 \times logP(LCS(c_1, c_2)) - (logP(c_1) + logP(c_2) \tag{2.5}$$

2.7.6 Lesk relatedness measure extended

This measure (Banerjee and Pedersen, 2003) takes into account the gloss overlap between the concepts And between the glosses of the hyponymic, hypertonic, meronymous and any another semantic relationship available in the thesaurus with the intuition that the more they overlap Its glosses, the more similar the concepts. This is summarized in the formula presented to continuation (2.6):

$$sim_{lesk}(c_1, c_2) = \Sigma_{r,q \in Relation(c_1,c_2)} Overlapping(gloss(r(c_1)), gloss(q(c_2))) \tag{2.6}$$

2.7.7 Wu and Palmer

he Wu and Palmer measure (wup) calculates relatedness by considering the depths of the two synsets in the WordNet taxonomies, along with the depth of the LCS. The formula is (2.7). This means that $0 < score <= 1$. The score can never be zero because the depth of the LCS is never zero (the depth of the root of a taxonomy is one). The score is one if the two input synsets are the same.

$$wup = 2 * depth(lcs)/(depth(s1) + depth(s2)) \tag{2.7}$$

2.7.8 Leacock and Chodorow

The relatedness measure proposed by Leacock and Chodorow is formula (2.8), where length is the length of the shortest path between the two synsets (using node-counting) and D is the maximum depth of the taxonomy.

$$lch = -log(length/(2 * D)) \tag{2.8}$$

The fact that the lch measure takes into account the depth of the taxonomy in which the synsets are found means that the behavior of the measure is profoundly affected by the presence or absence of a unique root node. If there is a unique root node, then there are only two taxonomies: one for nouns and one for verbs. All nouns, then, will be in the same taxonomy and all verbs will be in the same taxonomy. D for the noun taxonomy will be somewhere around 18, depending upon the version of WordNet, and for verbs, it will be 14. If the root node is not being used, however, then there are nine different noun taxonomies and over 560 different verb taxonomies, each with a different value for D.

If the root node is not being used, then it is possible for synsets to belong to more than one taxonomy. For example, the synset containing turtledove#n#2 belongs to two taxonomies: one rooted at group#n#1 and one rooted at entity#n#1. In such a case, the relatedness is computed by finding the LCS that results in the shortest path between the synsets. The value of D, then, is the maximum depth of the taxonomy in which the LCS is found. If the LCS belongs to more than one taxonomy, then the taxonomy with the greatest maximum depth is selected (i.e., the largest value for D).

2.7.9 Hirst and St-Onge

This measure (hso) works by finding lexical chains linking the two word senses. There are three classes of relations that are considered: extra-strong, strong, and medium-strong. The maximum relatedness score is 16.

2.8 Evaluation measures

There are some measures used to quantify the performance obtained in classification tasks. In this case, the classification task is to decide whether two input expressions are or not a true pair of paraphrases. Such measures are: accuracy, precision, recall and measurement F (F-measure) which are described below. Obtaining such measures is based on the counting of pairs classified correctly or incorrectly, which are:

- **TP (True Positives):** represents the number of correctly paired true pairs.
- **TN (True Negatives):** displays the number of false pairs correctly classified (ie, classified as false).
- **FP (False Positives):** shows the number of false pairs classified as true.
- **FN (False Negative):** represents the number of true pairs classified as negative.

2.8.1 Accuracy

Accuracy (accuracy) measures the percentage of correct total results obtained, so, If 50 pairs of 100 are correctly sorted, then their accuracy will be 50 . The formula (2.9): shows how to calculate this value.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.9}$$

2.8.2 Precision

This measure expresses the percentage of pairs classified as true that were actually true, That is, it measures how well the classification is relative to a specific label. Formula (2.10): shows how to calculate this measure.

$$accuracy = \frac{TP}{TP + FP} \tag{2.10}$$

2.8.3 Recall

The recall measure indicates which percentage of positive pairs were detected when performing the classification. Formula (2.11): shows how to calculate this measure.

$$accuracy = \frac{TP}{TP + FN} \tag{2.11}$$

2.8.4 Measure F1

The measure F (F-measure) is the harmonic mean between precision and recovery. The formula (2.12)

$$F_\beta = (1+\beta^2)\frac{presicion \times recall}{(\beta^2 \times precision) + recall} \tag{2.12}$$

although for the evaluation generally the symbol β is equal to 1, so the formula (2.12) used becomes (2.13)

$$F_1 = 2 \times \frac{presicion \times recall}{precision + recall} \tag{2.13}$$

Chapter 3

Methods for anaphora Resolution

3.1 pronoun resolution

3.1.1 supervised

3.1.1.1 Learning to Disambiguate Relative Pronouns

In this investigation is showed how a natural language system can find out to the previous of relative pronouns. it uses a well-known conceptual clustering system to create a case-based memory that prognosticate the antecedent of a wh-word given a information of the clause that come from. The automated approach duplicates the representation of hand-coded rules. Summing up, it needs only minimum syntactic parsing capabilities and a very general semantic quality for describing nouns. The manlike colaboration is needed during the practise phase.It is possible to compile relative pronoun disambiguation heuristics tuned to the syntactic and semantic preferences of a new control with relative ease. Furthermore, it considers that the manner of doing it , provide a general close for the automated acquisition of additional disambiguation heuristics for real language systems, especially for problems that need the adaptation of syntactic and semantic data.

3.1.2 unsupervised

3.1.2.1 Solving Hard Coreference Problems

The resolution of the Coreference is a problem of natural language comprehension from which reliable solutions can be obtained. A complication that arises is to solve cases that involve pronouns as they often require a deep understanding of language and the use of knowledge. In this paper is proposed an algorithmic solution that involves a new representation of the knowledge and it requires to deal with problems of coreference, together with a limited optimization framework that uses this knowledge in the decision making of corregencia. This representation is automatically compiled in the constraints that affect the decision of the coreference. Is presented a general coreference resolution system that notably improves the state of the art performance in hard, Winograd, and pronouns in cases of resolution, while still performing in the state-of-the-art coreference standard in joint data resolution.

3.1.2.2 An Example-Based Approach to Difficult Pronoun Resolution

The Winograd design shows identical phrases that contain a referential ambiguity which means that for a person it is easy to analyze but not for a computer. This work inspects the overall characteristics and knowledge required for such design. As you can see people tend to avoid ambiguous backgrounds when using written pronouns. This shows a method that acquires similar examples to the designs of Winograd but which have less ambiguity. A search is performed that captures the essential parts of a given font and then the alignments of the source sentence and their recovered examples are centered. Experimental results show that the sentence on the Web contains useful worlds for pronoun resolution.

3.2 Winograd scheme resolution

3.2.1 supervised

3.2.1.1 Combing Context and Commonsense Knowledge Through Neural Networks for Solving Winograd Schema Problems

This thesis proposes a general framework to combine the context and the common knowledge to solve the Winograd (WS) scheme and the problems of the pronoun disambiguation (PDP). Common sense knowledge bases are (eg pairs of cause-effect words) quantified as knowledge limitations. The limitations guide us to learn the improved embodiments of knowledge (KEE) of large text corpus. This framework resides in the pre-trained KEE models, proposes two methods to solve WS and PDP problems. The first method is an unsupervised method that represents all the pronouns and mentions of candidates in continuous vector spaces based on their contexts and calculates the semantic similarities between all pairs of acceptable words. The pronoun disambiguation procedure could then be implemented by comparing the semantic similarities between the pronoun (to be solved) and all mentions of the candidate. The second method is a supervised method, which extracts characteristics for all the pronouns and mentions of the candidate solves WS problems by training. Like the first method, the features used in the second method are also extracted based on the KEE models. Tests performed on the available PDP and WS test sets show that both achieve consistent improvements over basal systems. The best performance achieves 63 accuracy in the PDP test suite of the first Winograd Schema Challenge.

3.2.1.2 Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge

The task was to solve complex cases of definite pronouns, specifically those of traditional restrictions of the coreference (for example, binding restrictions, gender and number matching), as well as the commonly used resolution heuristics. Being able to solve this work has broader complications in artificial intelligence: a restricted version of it, sometimes called the challenge of the Winograd scheme, has suggested as a conceptual and practically attractive alternative to the Turing Test. A broad knowledge approach is used for this work, which produces a pronoun to be solved that exceeds the state-of-the-art resolvers by almost 18 points of accuracy in the data set.

3.2.2 knowledge

3.2.2.1 An Approach to Solve Winograd Schema Challenge Using Automatically Extracted Commonsense Knowledge

The Winograd Schema Challenge is an alternative to the Turing test. A Winograd scheme consists of sentences and questions such that the answer to the question depends on the resolution of a pronoun defined in the sentence. The answer is notorious for humans, but it is difficult for computers, as it requires knowledge of words or concepts in the sentence. In this article we propose a technique that analyzes the text semantically and seeks the common sense knowledge that is necessary to use the knowledge and answer a given question.

3.2.2.2 An Example-Based Approach to Difficult Pronoun Resolution

Dismayed by the Turing test's ability to properly evaluate whether a system exhibits human intelligence, the Winograd Scheme Challenge (WSC) It has been proposed as a solution. A Winograd scheme consists of a sentence and a question. The answers to the questions are intuitive for humans But they are designed to be complex toward machines, as they require various forms of common-sense knowledge about prayer.

It presents an approach that identifies known respondents to a challenge question, hunt down that knowledge of text repositories, and then reasons with them to come up with the answer. In the process a semantic analyzer was made (Www.kparser.org). It shows the approach It works well with respect to a subset of Winogradschemas

3.2.3 Inferences

3.2.3.1 Tackling the Winograd Schema Challenge Through Machine Logical Inferences

Levesque has said that the problem of solving difficult pronouns in a carefully chosen set of identical sentences defines it as the Winograd Schema Challenge (WSC), which could serve as a conceptual and practically attractive resource for the well-known Turing work. As he mentioned “anything that correctly answers a series of these questions is to think in the full sense as it is usually only for people. In this work they solve cases of definite pronouns. Specifically, those for which co-reference constraints as well as commonly used resolution heuristics are not useful, or the procedure they follow is very similar to a statistical approach, without involving the common logic as people do.

3.2.3.2 The Winograd Schema Challenge and Reasoning about Correlation

The Winograd Schema Challenge is a Turing test solution that can give a more meaningful measure of computer intelligence. This has a set of problems of resolution of the coreference that can not be solved without human thought. It is considered that the solution to such a problem lies in the coherence of discourse. Specifically, we examine two types of rhetorical relationships that can be used to establish discourse coherence: positive and negative correlation. It proposes a framework for reasoning on the correlation between sentences, and shows how this framework can be used to give solutions to some problems Winograd Schema

Chapter 4

Proposed method

The proposed method in this thesis tries to find automatically the correct antecedent in an anaphoric relation, the method is based in syntactic information which is obtained from the Winograd schemas, besides make use a resource file that contains the typical and more frequent concordance between Verbs and nouns, the method use to many similarity measures using the WordNet::Similarity [1]. y WordNet 3.0 [2].

The proposed method is not a supervised method because does not use semantic tagged statement to learn neither to build semantic classifiers, and this method neither is an unsupervised method because does not make a classification using big corpora. this method is knowledge-based because use information resources like a WordNet and Sketch Engine.

In general, this method chooses an antecedent considering similarity between nouns in the context and antecedents ,and relatedness based in concordance between candidates and verbs , this noun can be very specific so it necessary to get semantic relation like synonyms and hypernyms. so this semantic relation will form a set of new antecedents for each antecedent.

This set of antecedent, that have a semantic relation between them, will be the input for the antecedent selection algorithm that select the set of antecedent with highest score using semantic measures Pedersen et al. [77].

4.1 Description of proposed system

The system that was implemented is conformed by different modules that were built independently, so each module can be util for others task in natural language processing. in this first chapter stage it is explain each module, and in the following section it will explain inputs, functionality and outputs for each module within the system. the figure 4.1 shows this shows a blocks diagram with the interaction between modules within the system.

4.1.1 Antecedents, pronoun and solution extraction

This module read a XML file which have Winograd Schemes, the figure 4.2 show an example of a Winograd scheme in XML format, the module read a winograd schema and make Schema-object that is defined by definition 2:

Definition 2. *A* ***Schema-object*** S_i ***is a dictionary*** *with* $\langle text, P, C_1, C_2, ans \rangle$ *where*

- *text : text in the winograd schema.*
- *P : pronoun to disambiguate.*
- C_1 *: first antecedent in the anaporic relation*

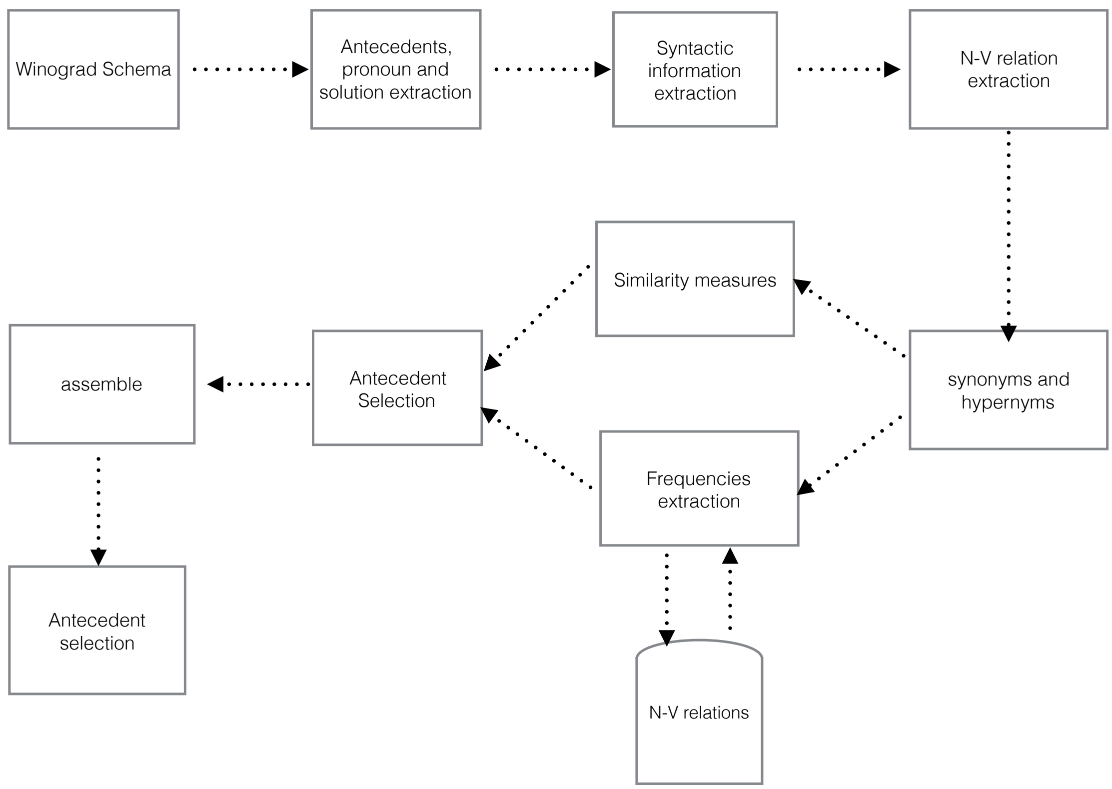

Figure 4.1: General system architecture

```xml
<schema>
    <text>
        <txt1>The bee landed on the flower because it had pollen.</txt1>
        <pron> it  </pron>
        <txt2> had pollen.</txt2>
    </text>
    <quote>
        <pron>it </pron>
        <quote2> had pollen.</quote2>
    </quote>
    <answers>
        <answer>the bee </answer>
        <answer>the flower </answer>
    </answers>
    <correctAnswer>A.  </correctAnswer>
</schema>
```

Figure 4.2: XML example of Winograd schema statement

- *C_2 : second antecedent in the anaporic relation*
- *ans : correct antecedent for the winograd schema*

The input for this module is a XML Winograd schema and the ouput is the Schema -object S_i

4.1.2 verb-noun concordance relation

This module get the most frequent concordance between verb and noun and their frequency that exist in sketch engine data base, it obtain a list of approximately one million of concordances. and create a file which store this concordances and their frequencies.

This module is based in the model TF-IDF (term frequency - inverse document frequency), this module calculate for each concordance relation stored in the concordance file this value, this model is used in classification task and document similarity task, representing each document by a vector and comparing multidimensionally. However for this task a document is equivalent to each concordance relation between verb and noun that exist in the concordance file, each concordance relation is defined with the follow definition: 3:

Definition 3. *A **verb-noun concordance relation** is a tuple with $\langle V, N, frecc\rangle$ where*

- *V is a verb and tuple head.*
- *N is a nonun and tuple modifier.*
- *$frecc$ is the numer of times that V and N are together in the sketch engine source.*

for a noun (or modifier in this thesis) is necessary calculate the normalized frequency and his inverse-frequency, inverse-frequency is the number of verbs that have concordance relation with this noun, and finally the concordance relation weight give the concordance relation importance between this noun and any verb, the normalized frequency is given by (4.1):

$$f_{v,n} = \frac{frecc_{v,n}}{maxfrecc_{1,n}} \tag{4.1}$$

where:

- $frecc_{v,n}$ is the concordance relation frecuency between $noun_i$ and $verb_j$.
- $maxfrecc_{1,n}$ is the maximun frecuency that have any noun with $verb_j$

The noun inverse - frequency in given by (4.2):

$$idf_i = log\frac{N}{n_i} \tag{4.2}$$

where:

- N is the total number of verb in the concordance source file.
- n_i is the number of verb that have concordance relation with the noun i.

Finally, the weight that had the concordance relation between verb and noun in given by: (4.3):

$$w_{i,j} = f_{i,j} \times idf_i \tag{4.3}$$

where:

- $f_{i,j}$ is the normalized frequency of the noun i with the verb j
- idf_i is the inverse-frequency of the noun i.

TF gives the noun importance regard the verb which is part or his concordance relation, so the weight increases if the noun is more frequent with this verb, otherwise, IDF gives the noun importance regard the all the verbs which is part or his concordance relation, so the noun weight decreases if this noun appears more frequently with the others verbs in the resource.

so the input for this module is a verb-noun concordance relation and the output is the follow tuple $\langle V, N, frecc, tf, idf, w\rangle$ where

- V is a verb and tuple head.
- N is a nonun and tuple modifier.
- $frecc$ is the numer of times that V and N are together in the sketch engine source.
- tf is the term-frequency obteined by (4.1).
- idf is the term-frequency obteined by (4.2).
- w is the wight of the noun obteined by (4.3).

An ouput exmple is showed in table 4.1

4.1.3 Syntactic parser

This module receives the text in the scheme - object defined by definition 2 and get syntactic tree using Stanford Parser [47], this module tagged each word in the statement and add this tag POS to the S_i and add all relation on dependency too.

For example:

$S_i(text) =$ *the bee landed in the flower because it want pollen.*

get tagging POS

$S_i(POS) =$ *the/DT bee/NN landed/VBD in/IN the/DT flower/NN because/IN it/PRP want/VBP pollen/NN*

Table 4.1: Example of tuples $\langle V, N, frecc, tf, idf, w\rangle$

Verb	**Noun**	**frequency**	**tf**	**idf**	**2**
take	place	185576	0.11822153	2.163086	0.25572333
be	part	164582	0.048840113	1.7947271	0.08765467
be	way	133611	0.039649393	1.6282218	0.06455801
be	time	127544	1.7805147E-6	1.4735535	2.6236837E-6
be	thing	83126	0.024667844	1.7624434	0.04347568
take	part	81055	0.051636234	1.7947271	0.092672944
be	member	77505	0.022999799	2.0418725	0.046962656
be	something	77092	0.02287724	2.2324471	0.05107223
be	case	74254	0.022035057	2.1579447	0.047550432
be	year	69744	1.4837623E-6	1.5949645	2.3665482E-6
be	problem	68683	0.02038185	2.0705914	0.042202484
be	place	62293	0.0184856	2.163086	0.03998594
be	number	61279	0.018184694	1.7596849	0.03199933
be	lot	59444	0.017640153	2.2357676	0.039439283
provide	information	58785	0.03744909	2.0086408	0.07522177
provide	service	58505	0.037270717	1.7982997	0.06702392
be	bit	58102	0.017241912	2.079086	0.035847418
be	nothing	51171	0.0151851205	2.8770673	0.043688614
have	time	51145	5.7334664E-6	1.4735535	8.44857E-6
make	decision	50900	0.032425936	2.5711317	0.083371356
be	people	50826	1.7805147E-6	1.4104064	2.5112492E-6
have	effect	50770	0.03234312	2.0464492	0.06618855
have	problem	49299	5.0964145E-6	2.0705914	1.0552592E-5
be	man	48795	0.014480037	1.625207	0.023533056
play	role	47274	0.030115988	2.337194	0.07038691

get dependencies relation

$S_i(parse) =$ *det(bee-2, the-1) , nsubj(landed-3, bee-2) , root(ROOT-0, landed-3) , case(flower-6, in-4) , det(flower-6, the-5) , nmod(landed-3, flower-6) , mark(want-9, because-7) , nsubj(want-9, it-8) , advcl(landed-3, want-9) , dobj(want-9, pollen-10)*

The input for this module is the text in the Schema-object and the output is the same Schema-object with POS and parse.

4.1.4 verb - noun bigrams

This modules have three method, all methods have the same input type and the same output type, the firs method return the bigrams that was obtained in the above section. the second method use of WordNet to get the sense for the word and all synonyms for each sense. finally the third method use WordNet [5] to obtain the first hypernyms

4.1.4.1 simple bigrams

This module receives syntactic bigrams of constituents, this bigrams are mixed bigrams because are formed by the grammatical category tag and the word in the sentences, so in this module return all the bigrams that are nouns and verbs because we are looking for the most frequently verb-noun concordance relation.

So the input for this module is a schema-object and the output is a list of mixed bigrams for this schema-object

For example:

input

$$S_i(text) = \textit{the bee landed in the flower because it want pollen.}$$

output

(bee/NN),
(landed/VBD),
(flower/NN),
(want/VBP),
(pollen/NN)

4.1.4.2 derivation of bigrams

This module receives syntactic bigrams of constituents, this bigrams are mixed bigrams because are formed by the grammatical category tag and the word in the sentences, so in this module return all the bigrams that are nouns and verbs because we are looking for the most frequently verb-noun concordance relation.

Additionally the method use WordNet to get the different senses of the word and get all the words in each synset and their synonyms.

So the input for this module is a schema-object and the output is a list of mixed bigrams and it synonyms for this schema-object.

For example:

input

$$S_i(text) = \textit{the bee landed in the flower because it want pollen.}$$

output

(bee/NN),
(landed/VBD),
(flower/NN),
(plant/NN),
(want/VBP),
(pollen/NN)

4.1.4.3 more general bigrams

This module receives syntactic bigrams of constituents, this bigrams are mixed bigrams because are formed by the grammatical category tag and the word in the sentences, so in this module return all the bigrams that are nouns and verbs because we are looking for the most frequently verb-noun concordance relation.

Additionally the method use WordNet to get the different senses of the word and get all the words in each synset and their synonyms.

Additionally the method use WordNet to get the hypernyms to the word and get all the words in each synset and their synonyms.

So the input for this module is a schema-object and the output is a list of mixed bigrams, its synonyms and its hypernyms for this schema-object.

For example:

input

$$S_i(text) = \textit{the bee landed in the flower because it want pollen.}$$

output

(bee/NN),
(animal/NN),
(insect/NN),
(landed/VBD),
(flower/NN),
(flora/NN),
(want/VBP),
(pollen/NN),
(seed-plant/NN)

4.1.5 Measures of semantic Relatedness

the method of semantic relatedness is a collection of six measures, Wu and Palmer [8], Jiang and Conrath [72], Leacock and Chodorow, [7] Lin [10], Resnik, [78] Path length [9], this measure were selected because there got the best result in the experiments, and because have the necessary complement to get a assemble algorithm together with the concordance resource.

The Jian-Conrath measure is based in Information Contente (IC) and path size between concepts. Unlike to Resnik, who introduce this concept, Jian-Conrath not only use Information Content that is proportioned by Least Common Subsumer (LCS) of both concepts, unless this use the information content proportions by each compared concept.

Lin is based in the same principle, he take the common information between the concepts like LCS and Information Content, this measure and the above are very similar, but they were developed by separated.

the adapted Lesk measure is a varian of the original Lesk [12] algorithm, which is based in gloss compression, and this adaptation is based in take the Neighbors's gloss making use of the hierarchy of Wordnet.

this module use the library "WordNet::Similarity" to perform each measure of similarity and make a dictionary with each worth with each antecedent.

4.2 Antecedent selection

In above sections, it was revised the process in which is obtained bigrams, tff-idf values and relatedness measures about a schema object S. this section describes the general process used to get the correct antecedent between two S_{c1} and S_{c2} in schema-object S, this antecedent are written in natural language and was used the follow methods.

Method 1: Direct antecedent was used with complementaries techniques in Natural Language Processing (NLP) like stemming, remove stop words, and negation to get the noun and verb, to find their frequency in the concordance resource file.

Method 2: Antecedent with the method 1 and all synonyms in all senses founded in WordNet was used to make a list of candidates antecedents to get the noun and verb, to find their frequency in the concordance resource file.

Method 3: Antecedent with the method 2 and hypernyms in all senses founded in WordNet was used to make a list of candidates antecedents t to get the noun and verb, to find their frequency in the concordance resource file.

Method 4: Antecedent with the method 3 was used to find the minimum tf-idf worth and this antecedent was be selected.

Method 5: Antecedent with the method 1 and all constituents bigrams was used to get all measures of relatedness by each bigram and candidate, and the minimum measure is selected.

Follow sections describe the above method used to resolve the anaphoric antecedent.

4.2.1 simple method

The first method used to resolve the anaphoric antecedent is called simple method, because this get the mixed constituent bigrams that was generated by syntactic parser module return a constituent bigram list. An ouput exmple is showed in table 4.2

Table 4.2: List mixed constitued bigrams taht was obtained in the syntactyc parser module for the $S_i(text)$ = *the bee landed in the flower because it want pollen.*

mixed constituent bigrams
(NN, bee)
(VBD, landed)
(NN, flower)
(VBP, want)
(NN, pollen)

it is important say that the module realized a step in which some POS tag was was generalized, with this step obtains more bigrams. this generalization stop was applied on verbs and nouns, because this have different variants like substantive categories like plurals (NNS), proper names (NNP) and plurals proper names (NNPS), so this categories were generalize to (NN), the same way for verbs because these have different categories like past ver (VBD), gerund verb (VBG), participle verb (VBN) and third person verbs (VBZ) and these were generalized to VB. An exmple is showed in table 4.3

Table 4.3: List mixed constitued bigrams taht was obtained in the syntactyc parser module and his generalization

mixed constituent bigrams	**mixed stemmed constituent bigrams**
(NN, bee)	–
(VBD, landed)	(VB, landed)
(NN, flower)	–
(VBP, want)	(VB, want)
(NN, pollen	–

4.2.1.1 stemming

In the chapter 2 was mentioned the stemming task, that is a process that remove the suffix and prefix of the word because this does not give relevant information, for example Applying to the words **dancer**, **dancing** and **danced** the process get **dance** for all the words.

the stemming process used the porter algorithm [79]. the system use stemming because the **concordance source file** have noun and verb in stem form. This process is necessary to get correctly the concordance relations.

this stemming process is applied to the **mixed constituent bigrams** this process is applied only to the word and the POS tag not change. An exmple is showed in table 4.4

Table 4.4: List mixed constitued bigrams taht was obtained in the syntactyc parser module for the $S_i(text)$ = *the bee landed in the flower because it want pollen.*

mixed constituent bigrams	**mixed stemmed constituent bigrams**
(NN, bee)	–
(VBD, landed)	(VB, land)
(NN, flower)	–
(VBP, want)	–
(NN, pollen	–

4.2.1.2 remove determinant

This task was added because both antecedents in the Winograd schema are the same type (male, female) or are both are plural or singular, for this reason the determinant not contain any information to helps anaphora resolution.

This module affect directly to the antecedents and the answer elements in the schema-object S_i so the information change, for example. $S_i(text)$: the bee landed on the flower because it want polle.

- $S_i(C_1)$ **the bee** change to **bee**.
- $S_i(C_2)$ **the flower** change to **flower**.
- $S_i(ans)$ **the bee** change to **bee**.

4.2.1.3 simple-verb-noun frequency

to get verb-noun frequency the simple method make a list of tuples with each antecedent, all verbs in the constituens bigramas, and the frequency of their, this three elementns this tuple is defined by **Definition 4**.

Definition 4. *A* ***tulpe*** T_i ***is a tuple*** *with* $\langle V, N, f\rangle$ *where*

- V *: verb in the constituents bigrams.*
- N *: noun or antecedent.*
- f *: number of time that the* V *and* N *appears together in the concordance resource file*

with this list the algorithm 4.1 compare each frequency by antecedent and get the antecedent with the max frequency.

Algorithm 4.1: simple algorithm

$A \leftarrow null$, $V \leftarrow listOfVerbs$, $C_1 = S_i(C_1)$, $C_2 = S_i(C_2)$, $Answer = S_i(ans)$
for each $verb \subseteq V$ **then**:
  if $freq(verb, C_1) > freq(verb, C_2)$ **then**:
    $A \leftarrow C_1$
  end if
  else $freq(verb, C_2) > freq(verb, C_1)$ **then**:
    $A \leftarrow C_2$
  end else
end for
if $Answer == A$ **then**:
  return 1
end if
else:
  return 0
end else

This algorithm no ever find a result, because the nouns are very specific, so the follow methods get more general nouns.

4.2.2 derivation method

the second method that is used to resolve the anaphoric antecedent is called derivation method, because this get the mixed constituent bigrams that was obtained by syntactic parser module that return a constituent list, apply that above method mentioned (stemming and remove determinant).

With the mixed method list, get a list for each antecedent and using Wordnet get their synonyms. in this way the method can get more frequencies to disambiguate the pronoun.

With this major list of nouns the algorithm 4.2 compare each frequency by antecedent and his derivation and get the antecedent with the max frequency.

Algorithm 4.2: derivation algorithm

$A \leftarrow null$, $V \leftarrow listOfVerbs$, , $C_1 = S_i(C_1)$, $C_2 = S_i(C_2)$, $L_{c_1} \leftarrow getSynonyms(C_1)$, $L_{c_2} \leftarrow getSynonyms(C_2)$ $Answer = S_i(ans)$
for each $verb \subseteq V$ **then**:
  for each $A_{C_1} \subseteq L_{c_1}$ **then**:
    for each $A_{C_2} \subseteq L_{c_2}$ **then**:
      if $freq(verb, A_{C_1}) > freq(verb, A_{C_2})$ **then**:
        $A \leftarrow A_{C_1}$
      end if
      else $freq(verb, A_{C_2}) > freq(verb, A_{C_1})$ **then**:
        $A \leftarrow A_{C_2}$
      end else
    end for
  end for
end for
if $Answer == A$ **then**:
  return 1
end if
else:
  return 0
end else

This algorithm no ever find a result, because the nouns and her synonyms are very specific, so the follow methods get more general nouns.

4.2.3 Hypernyms method

The third method that is used to resolve the anaphoric antecedent is called hypernyms method, because this get the mixed constituent bigrams that was obtained by syntactic parser module that return a constituent list, apply that above method mentioned (stemming and remove determinant).

With the mixed method list, get a list for each antecedent and using WordNet get their synonyms and hypernyms. in this way the method can get more general frequencies to disambiguate the pronoun.

With this major list of nouns the algorithm 4.3 compare each frequency by antecedent and his derivation and get the antecedent with the max frequency.

Algorithm 4.3: hypernyms algorithm

$A \leftarrow null$, $V \leftarrow listOfVerbs$, , $C_1 = S_i(C_1)$, $C_2 = S_i(C_2)$, $L_{c_1} \leftarrow getHypernyms(C_1)$, $L_{c_2} \leftarrow getHypernyms(C_2)$ $Answer = S_i(ans)$
for each $verb \subseteq V$ **then**:
  for each $A_{C_1} \subseteq L_{c_1}$ **then**:
    for each $A_{C_2} \subseteq L_{c_2}$ **then**:
      if $freq(verb, A_{C_1}) > freq(verb, A_{C_2})$ **then**:
        $A \leftarrow A_{C_1}$
      end if
      else $freq(verb, A_{C_2}) > freq(verb, A_{C_1})$ **then**:
        $A \leftarrow A_{C_2}$
      end else
    end for
  end for
end for
if $Answer == A$ **then**:
  return 1
end if
else:
  return 0
end else

This algorithm no ever find a result, because there are nouns that don‘’t have a frequency.

4.2.4 tf-idf method

This method is based in the model TF-IDF (term frequency - inverse document frequency), this module calculate for each concordance relation stored in the concordance file this value, this model is used in classification task and document similarity task, representing each document by a vector and comparing multidimensionally. However for this task a document is equivalent to each concordance relation between verb and noun that exist in the concordance file

this method use the same list like a third method but don't use the frequency values, this method use the weight W that was obtained in the module tf-idf in a section above.

Similarly with the above methods, the algorithm 4.4 compare each verb by antecedent and get the weight value.

Algorithm 4.4: tf-idf algorithm

$A \leftarrow null$, $V \leftarrow listOfVerbs$, $C_1 = S_i(C_1)$, $C_2 = S_i(C_2)$, $Answer = S_i(ans)$
for each $verb \subseteq V$ **then**:
    if $w(verb, C_1) > w(verb, C_2)$ **then**:
        $A \leftarrow C_1$
    end if
    else $w(verb, C_2) > w(verb, C_1)$ **then**:
        $A \leftarrow C_2$
    end else
end for
if $Answer == A$ **then**:
    return 1
end if
else:
    return 0
end else

This algorithm no ever find a result, because there are nouns that don''t have a frequency.

4.2.5 Measures method

The method number five use to resolve the anaphoric antecedent the measure semantic relatedness. Because the concordance resource is limit, and don't have all the concordance relation.

But the most important is the parameter that measure receive, this must be of the same POS tag, that is to say, that a difference with the above method that need a ver and a noun, this method need (N,N) or (V,V), int this case this method only use the Nouns.

This method get a list with all nouns after the pronoun to resolve in the, this nouns goin to be the context and the input for the measures.

Similarly with the above methods, the algorithm 4.5 compare each noun by antecedent and get the measure worth.

The max value will be the antecedent for the pronoun that are in anaphoric relation.

Algorithm 4.5: measure algorithm

$A \leftarrow null$, $L_n \leftarrow listOfNouns$, $C_1 = S_i(C_1)$, $C_2 = S_i(C_2)$, $Answer = S_i(ans)$
for each $noun \subseteq L_n$ **then**:
    if $getMeasures(noun, C_1) > getMeasures(noun, C_2)$ **then**:
        $A \leftarrow C_1$
    end if
    else $getMeasures(noun, C_2) > getMeasures(noun, C_1)$ **then**:
        $A \leftarrow C_2$
    end else
end for
if $Answer == A$ **then**:
    return 1
end if
else:
    return 0
end else

the method of semantic relatedness is a collection of six measures, Wu and Palmer [8], Jiang and Conrath [72], Leacock and Chodorow, [7] Lin [10], Resnik, [78] Path length [9], this measure were selected because there got the best result in the experiments, and because have the necessary complement to get a assemble algorithm together with the concordance resource.

4.3 Assemble algorithm

In this thesis is used an assemble algorithm because when there are many decision to take. is better count with a expert, because he has the better solution, this thesis have 10 experts. the experts are classified by follow:

frecuency and semantic Verb-Noun experts

- **frequency of verb and noun together:** this experts is who get the number of times in which a verb appears together with something noun
- **frecuency of verb and similar noun together:** this expert is who get the number of times in which a verb appears together not only with a simple noun, this expert consider the semantic relation of synonymy.
- **frecuency of verb and general noun together:** this expert is who get the number of times in which a verb appears together not only with a simple noun, this expert consider the semantic relation of synonymy and generalization getting the semantic relation of hyperonymy.
- **tf-idf of a verb and noun together:** this expert is who get the weight of the relation between a verb and a noun, with considering all the frequencies and relations. This expert consider the semantic relation of synonymy and generalization getting the semantic relation of hyperonymy.

similarity relatedness experts

they are six experts that can give a numeric value that quantifies how similar or related two concepts are, being able to organize concepts by there similarity ir relatedness to each other. this is a fundamental operation in the human mind, and this experts give it.

for this reason this thesis using assemble algorithm 4.6 with this methods and measures to de a rotation about what antecedent is more relatedness with the context.

Algorithm 4.6: assemble algorithm

$C_1 \leftarrow 0$, $C_2 \leftarrow 0$, $L_{values} \leftarrow getAntecedents()$
for each $candidate \subseteq L_{values}$ **then**:
    if $candidate == C_1$ **then**:
        $C_1 \leftarrow C_1 + 1$
    end if
    else if $candidate == C_2$ **then**:
        $C_1 \leftarrow C_1 + 1$
    end else if
return $max(C_1, C_2)$

Chapter 5

Results

In this chapter there are three section, in the first are defined the evaluation metrics like recall, precision and F1, they are using to validate the obtained results by the method proposed. In the second section is the results, tables and graphs. Finally there is a discussion about the results.

5.1 Evaluation metrics

To evaluate the proposed method, there are three metrics that are using in the tasks of Word Sense Disambiguation, resolution of coreference and resolution of anaphora. this metric are precision, recall and F1.

anaphora resolution tray to find the correct antecedent for a noun or in this case pronoun, if the resolution method not give a result because it can't determinate which antecedent is, and this antecedent is determinate as null, so *precision* evaluates all antecedent that have a value. and *recall* considers all values although this value be null. for this reason *precision* is mayor or equal to *recall*

is important tha a method can identify the antecedents that can't identify, because those can be identified by other method with math *precision*

the precision is given by (5.1):

$$precision = \frac{c}{c+i} \tag{5.1}$$

where:

- c is the number of correct antecedent given by the system.
- i is the number of incorrect antecedent given by the system.

another formula for precision is given by (5.2):

$$precision = \frac{TP}{TP+FP} \tag{5.2}$$

where:

- TP (True Positives): represents the number of correctly paired true pairs.
- FP (False Positives): shows the number of false pairs classified as true.

the recall is given by (5.3):

$$precision = \frac{c}{c + i + n} \tag{5.3}$$

where:

- c is the number of correct antecedent given by the system.
- i is the number of incorrect antecedent given by the system.
- n is the number of antecedent that not have value equals to null.

another formula for recall is given by (5.4):

$$precision = \frac{TP}{TP + FN} \tag{5.4}$$

where:

- TP (True Positives): represents the number of correctly paired true pairs.
- FN FN (False Negative): represents the number of true pairs classified as negative.

5.2 Description of experiments

In this section is showed and explain the characteristics that have each experiment and the obtained results.

5.2.1 Measures

the table 5.1 contains the measures used and its key for identification used in this thesis

Table 5.1: List of the measures used in this thesis and its key

key	name of mmeasure
s	simple measure
d	derivation measure
h	hypenymns measure
1	wup measure
2	jco measure
3	lch measure
4	lin measure
5	res measure
6	path measure
w	weigth measure

where:

- *simple measure* this measure is based in the original verbs and noun o the sentence and its frequency.
- *derivation measure* this measure is based in the synonyms of the verbs and noun o the sentence and its frequency.

- *simple measure* this measure is based in the hypernyms verbs and noun o the sentence and its frequency.
- *wup measure* this the Wu and Panlmer measure is based in depths of the two synsets in the WordNet taxonomies.
- *jco measure* this the Jiang and Conrath measure is based in the information content between the two synsets and the least common subsumer.
- *lch measure* this the Leacock and Chodorow measure is based in the size path between the two synsets and the deep of taxonomy.
- *lin measure* this the Lin measure is based in the information content between the two synsets and the least common subsumer.
- *res measure* this the Resnik measure is based in the information content of the least common subsumer.
- *path measure* this the Path measure is based in counting node between two synsets.
- *weigth measure* this measure is based in the hypernyms verbs and noun o the sentence and its relation relevance.

5.2.2 Corpus

The corpus is principal factor to get in mind, many corpus have POS tagged, class tagged, and WordNet tagged, this thesis only use one corpus this is the official corpus for the Winograd Scheme Challenge this corpus is divided into two files, one of them is the train file, and the another one is the test file.

This thesis use the test file, it contains 561 Winograd schemes with the correct antecedent tagged.

5.3 Results

The results are expressed in terms of accuracy, which is the percentage of correctly resolved the antecedent pronoun.

the result are showed in two section, the first section show the results in terms of accuracy, precision, recall, and F1, the second section show the result after the assemble method in the same term.

5.3.1 Results by measures

the section shows tables of result by each measure to identify which measure is better

the table 5.2 contains the accuracy value for each measure

the table 5.3 contains the precision value for each measure

the table 5.4 contains the recall value for each measure

the table 5.5 contains the F1 value for each measure

Table 5.2: accuracy value for each measure

key	accuracy
w	49.78
s	49.65
2-jco	49.65
3-lch	49.43
h	49.36
d	49.32
6-path	48.96
1-wup	47.80
5-res	45.81
4-lin	44.89

Table 5.3: Precision value for each measure

key	Precision
w	50.41
d	50
h	50
s	49.73
2-jco	49.73
3-lch	49.73
6-path	48.64
1-wup	48.01
5-res	45.45
4-lin	42

Table 5.4: recall value for each measure

key	recall
h	52.08
4-lin	51.12
w	50.83
s	49.77
d	48.44
2-jco	43.37
3-lch	42.53
1-wup	42.17
6-path	41.66
5-res	40

5.3.2 Assemble result

the section shows a table by each metric, each table contains 10 assemble measure

the firs table 5.6 show the accuracy assemble measure, Accuracy get the percentage of correct total results obtained

the second table 5.7 show the precision assemble measure that expresses the percentage of pairs classified as true that were actually true

Table 5.5: F1 value for each measure

key	F1
h	51.02
w	50.62
s	49.77
d	49.20
2-jco	46.34
3-lch	45.85
1-wup	44.90
6-path	44.88
5-res	42.55
4-lin	40

Table 5.6: accuracy value for assamble measure

key	accuracy
[1, r, s, 3, d, 4]	51.98
[1, r, s, 3, d, 4, 6, h]	51.78
[1, 2, r, s, d, 4, 6, h]	51.77
[1, r, s, d, 4, 6]	51.73
[2, r, s, 3, 6, h]	51.69

Table 5.7: precision value for assamble measure

key	precision
[1, s, 3, d, h]	52.29
[1, r, s, 3, d]	52.29
[1, r, s, 6]	52.28
[2, r, 3, d, 4, 6, h]	52.28
[1, 2, r, s, 4, h]	52.27

the third table 5.8 show the recall assemble measure that indicates which percentage of positive pairs were detected

Table 5.8: recall value for assamble measure

key	recall
[r, 5, h]	52.08
[h]	52.08
[r, 4, h]	52.08
[1, r, s, 5, h]	52.05
[1, r, h]	51.66

the fourth table 5.9 show the F1 assemble measure that is the harmonic mean between precision and recall.

Table 5.9: F1 value for assamble measure

key	**F1**
[s, 3, d, 6, h]	51.36
[r, s, 3, d, 6]	51.36
[1, r, s, 3, d, 6, h]	51.52
[1, r, s, 5, h]	51.46
[1, r, s, 6, h]	51.42

5.4 Comparison with previous methods

The table 5.10 presents a final position with the best values in the winograd schema challenge.

Table 5.10: accuracy values in the winograd schema challenge

autor	**name**	accuracy
(Rahman and Ng et al., 2012)	Narrative Chains	30.67
(Rahman and Ng et al., 2012)	Google	33.16
(Lee et al., 2011)	The Stanford resolver.	40.1
(quian Liu at al., 2016)	USSM	48.7
(quian Liu at al., 2016)	NKAM	49.1
-	The Random baseline.	50
(quian Liu at al., 2016)	USSM + NKAM	50.2
(Durrett and Klein, 2013)	BERKELEYnew	50.32
(Chang et al.2013),	Illinois	51.48
-	**our system**	**51.98**
(Canasai Kruengkrai, 2014)	-	69.68
(Rahman and Ng et al., 2012)	MENTRANKER	73.05
(Peng and Khashabi, 2015)	KnowComb	76.76

5.5 Discussion

The results show that is a challenge for the estate of art and in this thesis is proposed a new knowledge based technique to address this task, but there are still a large area to improve. its possible that the results will improve if is used machine learning techniques, knowledge from corpora and sentiment analysis. the independent result shown that a measure that use the relation between the verb of the pronoun and all the synonyms and hypernyms of a word, and getting its relevance of this relations, work better than the others, in this thesis the relevance was obtained by tf-idf that shown the relevance of a verb-noun relation between all relation in a corpus. this achieve an accuracy of 49.78 and a F1 measure of 50.02 this measure overpass some of the baseline proposed the semantic relatedness that obtained the two betters result was the measure proposed by Jiang and Conrath measure and Leacock and Chodorow measure achievement a recall of 49.65 and 49.43respectively and a F1 of 46.34 and 45.85 respectively. so the best relatedness for this task in this thesis are the measures that have in mind the information content between the two concepts and the least common subsumer. and the based in the size path between the two concepts and the deep of taxonomy The best accuracy in this thesis was achievement by the assemble method that is conformed by the simple, synonyms weight and the measure of relented based in taxonomy like a Wu Palmer, Leacock and Chodorow and Lin measure than have in mind the information content about there concepts and the last common subsumer. The best F1 in this thesis was achievement by the assemble method that is conformed by the simple, ,synonyms and hypernyms, and the measure of relented based in the path like a Leacock and Chodorow and Path measure this thesis is based in the importance of POS Tagged and used the two elements more important in a statement by separate, this elements are the noun that have the context of the sentence, this context is evaluate with the measure of relatedness, in the classic methods of disambiguation word sense is used many context that is conformed by more of 40 or 80 word, but in this case the schema don't have the noun sufficient to get a good context, so this deficient is taking by the other important element in a statement, verbs, this take this element using the relevance of a relation with a noun using the frequency of this in a specific corpus.

Chapter 6

Conclusions and future work

the resolution of anaphora problem in especial case the Winograd Scheme Challenge consist in to find the correct antecedent for a given pronoun that have a two candidates that a had the same type. there are many works that try to do this task with method of machine learning and reasoning in this thesis we use syntactic information, knowledge from WordNet and SketchEngine relation of verb and noun. and relatedness between words.

6.1 Conclusion

In this thesis we investigate the resolution of anaphora that is present in pronouns, this task is known as resolution of complex pronouns, this problem has a extensive discussion for researchers since 1970, but has had a new interest because this problem is seen like an another way or a substitution for the Turing Test. in this thesis was obtained the follow conclusions.

- taking a simple method we can achievement many of baseline proposed using the method of weight that method get the relevance between verb and noun taking the frequencies of this relation and the frequency of each word into the corpus
- measures that have in mind the information content between the two concepts and the least common subsumer, and the based in the size path between the two concepts and the deep of taxonomy have a better accuracy
- the semantic relation like synonyms and hypernyms help to improve the performance. this is because if a some word is not in the corpus of relatedness, is problaly that its synonyms appears. hypernyms give less information about relents but helps to find those nouns that are very specific.
- The asamble method merge two concepts, the concepts of relevance of verb and noun and the relatedness between nouns, because, the winograd scheme don't have the enough context to used in classic methods of coreference resolution.

6.2 Contributions

The principal contribution of this work are:

- A corpus with the relevance relation between verb and noun.
- Development and implementation of method to get the relevance between verb and noun.

- Development and implementation of a method to get the relevance between a noun and its derivation and a verb.
- development and implementation of a method to prepossessing (steaming, remove determinant) noun and get its semantic relation like a synonyms and hypernyms.
- A novel assemble algorithm to anaphora resolution based in kownoledge-rich techniques.
- Development and implementation of a method to do the power set with the different techniques and combine this results i the assemble method.

6.3 Future Work

As future work:

- To try with another corpus that there are in the Winograd Scheme Callenge, like a WNCorpus and PDPCorpus
- To implement this work in another corpus for the coreferences resolution like MUC-6, OntoNotes, etc.
- To try to implement a learning method to used the class label in the corpus.
- To try to implement methods of sentiment analysis, and causal references by verbs and reasoning.
- To use a mehtods of entity and person resolution.

Bibliography

[1] Rennie, J. (2000). *"WordNet:: QueryData: a Perl module for accessing the WordNet database"*. Retrieved July, 18, 2006.

[2] Beckwith, R., Fellbaum, C., Gross, D., Miller, K., Miller, G. A., and Tengi, R. (1990). *"Five papers on WordNet"*. Special Issue of Journal of Lexicography, 3(4), 235-312.

[3] Patwardhan, S., Banerjee, S., and Pedersen, T. (2003, February). *"Using measures of semantic relatedness for word sense disambiguation."* In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 241-257). Springer Berlin Heidelberg.

[4] De Marneffe, M. C., MacCartney, B., and Manning, C. D. (2006, May). *"Generating typed dependency parses from phrase structure parses"*. In Proceedings of LREC (Vol. 6, No. 2006, pp. 449-454).

[5] Fellbaum, C. (1998). *"WordNet: An electronic database"*.

[6] Jiang, J. J., and Conrath, D. W. (1997). *"Semantic similarity based on corpus statistics and lexical taxonomy"*. arXiv preprint cmp-lg/9709008.

[7] Leacock, C., and Chodorow, M. (1998). *"Combining local context and WordNet similarity for word sense identification. WordNet: An electronic lexical database"*, 49(2), 265-283.

[8] Wu, Z., and Palmer, M. (1994, June). *"Verbs semantics and lexical selection"*. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics (pp. 133-138). Association for Computational Linguistics.

[9] Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004, May). *"WordNet:: Similarity: measuring the relatedness of concepts"*. In Demonstration papers at HLT-NAACL 2004 (pp. 38-41). Association for Computational Linguistics.

[10] Lin, D. (1997, July).*" Using syntactic dependency as local context to resolve word sense ambiguity"*. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (pp. 64-71). Association for Computational Linguistics.

[11] Resnik, P. (1995). *"Using information content to evaluate semantic similarity in a taxonomy"*. arXiv preprint cmp-lg/9511007.

[12] Lesk, M. (1986, June). *"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.

[13] Porter, M. F. (1980). *" An algorithm for suffix stripping"*. Program, 14(3), 130-137.

[14] Shane Bergsma and Dekang Lin. 2006. Bootstrapping path-based pronoun resolution. In Proceedings of the 21st International Conference on Computational Lin- guistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 33–40.

[15] Volha Bryl, Claudio Guiliano, Luciano Serafini, and Kateryna Tymoshenko. 2010. Using background knowledge to support coreference resolution. In Pro- ceedings of the 19th European Conference on Artificial Intelligence, pages 759–764.

[16] Nathanael Chambers and Dan Jurafsky. 2008. Unsu- pervised learning of narrative event chains. In Pro- ceedings of the 46th Annual Meeting of the Associa- tion for Computational Linguistics: Human Language Technologies, pages 787–797.

[17] Donna K. Byron. 2002. Resolving pronominal reference to abstract entities. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguis- tics, pages 80–87.

[18] Eugene Charniak and Micha Elsner. 2009. EM works for pronoun anaphora resolution. In Proceedings of the 12th Conference of the European Chapter of the Asso- ciation for Computational Linguistics, pages 148–156.

[19] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed de- pendency parses from phrase structure parses. In Pro- ceedings of the 5th International Conference on Lan- guage Resources and Evaluation, pages 449–454.

[20] Hector J. Levesque. 2011. The Winograd Schema Chal- lenge. In AAAI Spring Symposium: Logical Formal- izations of Commonsense Reasoning.

[21] Bing Liu. 2012. Sentiment Analysis and Opinion Min- ing. Morgan and Claypool Publishers.

[22] Ruslan Mitkov, Richard Evans, and Constantin Orasan. 2002. A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In Al. Gelbukh, editor, Computational Linguistics and Intel- ligent Text Processing, pages 169–187. Springer.

[23] Rahman, A., and Ng, V. (2012, July). Resolving complex cases of definite pronouns: the winograd schema challenge. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 777-789). Association for Computational Linguistics.

[24] Aone, C., and Bennett, S. W. (1995, June). Evaluating automated and manual acquisition of anaphora resolution strategies. In Proceedings of the 33rd annual meeting on Association for Computational Linguistics (pp. 122-129). Association for Computational Linguistics.

[25] Banerjee, S., and Pedersen, T. (2002, February). An adapted Lesk algorithm for word sense disambiguation using WordNet. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 136-145). Springer Berlin Heidelberg.

[26] Bergsma, S. (2005, May). Automatic acquisition of gender information for anaphora resolution. In Conference of the Canadian Society for Computational Studies of Intelligence (pp. 342-353). Springer Berlin Heidelberg.

[27] Ge, N., Hale, J., and Charniak, E. (1998, August). A statistical approach to anaphora resolution. In Proceedings of the sixth workshop on very large corpora (Vol. 71, p. 76).

[28] Hoste, V. (2005). Optimization issues in machine learning of coreference resolution.

[29] Jiang, J. J., and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008.

[30] Ng, H. T., Zhou, Y., Dale, R., and Gardiner, M. (2005, July). A machine learning approach to identification and resolution of one-anaphora. In INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE (Vol. 19, p. 1105). LAWRENCE ERLBAUM ASSOCIATES LTD.

[31] Levesque, H. J. (2014). On our best behaviour. Artificial Intelligence, 212, 27-35.

[32] Sharma, A., Vo, N. H., Gaur, S., and Baral, C. (2015, March). An approach to solve winograd schema challenge using automatically extracted commonsense knowledge. In 2015 AAAI Spring Symposium Series.

[33] De Marneffe, M. C., and Manning, C. D. (2008). Stanford typed dependencies manual (pp. 338-345). Technical report, Stanford University.

[34] De Marneffe, M. C., MacCartney, B., and Manning, C. D. (2006, May). Generating typed dependency parses from phrase structure parses. In Proceedings of LREC (Vol. 6, No. 2006, pp. 449-454).

[35] Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41.

[36] Schüller, P. (2014, May). Tackling winograd schemas by formalizing relevance theory in knowledge graphs. In Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning.

[37] Liu, Q., Jiang, H., Ling, Z. H., Zhu, X., Wei, S., and Hu, Y. (2016). Combing Context and Commonsense Knowledge Through Neural Networks for Solving Winograd Schema Problems. arXiv preprint arXiv:1611.04146.

[38] Ng, V. (2010, July). Supervised noun phrase coreference research: The first fifteen years. In Proceedings of the 48th annual meeting of the association for computational linguistics (pp. 1396-1411). Association for Computational Linguistics.

[39] Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., ... and Franchini, M. et alia, 2011. Ontonotes release, 4.

[40] Cardie, C. (1992, July). Learning to disambiguate relative pronouns. In AAAI (pp. 38-43).

[41] Chinchor, N., Lewis, D. D., and Hirschman, L. (1993). Evaluating message understanding systems: an analysis of the third message understanding conference (MUC-3). Computational linguistics, 19(3), 409-449.

[42] Sharma, A., Vo, N. H., Gaur, S., and Baral, C. (2015, March). An approach to solve winograd schema challenge using automatically extracted commonsense knowledge. In 2015 AAAI Spring Symposium Series.

[43] Rahman, A., and Ng, V. (2012, July). Resolving complex cases of definite pronouns: the winograd schema challenge. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (pp. 777-789). Association for Computational Linguistics.

[44] Sharma, A. (2014). Solving winograd schema challenge: Using semantic parsing, automatic knowledge acquisition and logical reasoning (Doctoral dissertation, Arizona State University).

[45] Peng, H., Khashabi, D., and Roth, D. (2015). Solving hard coreference problems. Urbana, 51, 61801.

[46] Sharma, A., Vo, N. H., Aditya, S., and Baral, C. (2015, June). Towards addressing the winograd schema challenge-building and using a semantic parser and a knowledge hunting module. In Proceedings of Twenty-Fourth International Joint Conference on Artificial Intelligence. AAAI.

[47] De Marneffe, M. C., MacCartney, B., and Manning, C. D. (2006, May). Generating typed dependency parses from phrase structure parses. In Proceedings of LREC (Vol. 6, No. 2006, pp. 449-454).

[48] De Marneffe, M. C., and Manning, C. D. (2008, August). The Stanford typed dependencies representation. In Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation (pp. 1-8). Association for Computational Linguistics.

[49] Chambers, N., and Jurafsky, D. (2008, June). Unsupervised Learning of Narrative Event Chains. In ACL (Vol. 94305, pp. 789-797).

[50] Bos, J., and Markert, K. (2005, October). Recognising textual entailment with logical inference. In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing (pp. 628-635). Association for Computational Linguistics.

[51] Hobbs, J. R. (1978). Resolving pronoun references. Lingua, 44(4), 311-338.

[52] Bailey, D., Harrison, A., Lierler, Y., Lifschitz, V., and Michael, J. (2015). The Winograd Schema Challenge and Reasoning about Correlation. In In Working Notes of the Symposium on Logical Formalizations of Commonsense Reasoning.

[53] Peng, H., Khashabi, D., and Roth, D. (2015). Solving hard coreference problems. Urbana, 51, 61801.

[54] Bansal, M., and Klein, D. (2012, July). Coreference semantics from web features. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1 (pp. 389-398). Association for Computational Linguistics.

[55] Connolly, D., Burger, J. D., and Day, D. S. (1997). A machine learning approach to anaphoric reference. In New Methods in Language Processing (pp. 133-144).

[56] Ng, V., and Cardie, C. (2002, August). Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In Proceedings of the 19th international conference on Computational linguistics-Volume 1 (pp. 1-7). Association for Computational Linguistics.

[57] Ng, V., and Cardie, C. (2002, July). Improving machine learning approaches to coreference resolution. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (pp. 104-111). Association for Computational Linguistics.

[58] Ng, V. (2004, July). Learning noun phrase anaphoricity to improve coreference resolution: Issues in representation and optimization. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (p. 151). Association for Computational Linguistics.

[59] Rahman, A., and Ng, V. (2011, June). Coreference resolution with world knowledge. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 814-824). Association for Computational Linguistics.

[60] Peng, H., Khashabi, D., and Roth, D. (2015). Solving hard coreference problems. Urbana, 51, 61801.

[61] Ackerman, E. (2014). Can Winograd Schemas Replace Turing Test for Defining Human-Level AI?. IEEE Spectrum, July, 29.

[62] Davis, E., Morgenstern, L., and Ortiz, C. (2016). Human tests of materials for the Winograd Schema Challenge 2016.

[63] Morgenstern, L., Davis, E., and Ortiz Jr, C. L. (2016). Planning, Executing, and Evaluating the Winograd Schema Challenge. AI Magazine, 37(1), 50-54.

[64] Winograd, T. (1972). Understanding natural language. Cognitive psychology, 3(1), 1-191.

[65] Kruengkrai, C., Inoue, N., Sugiura, J., and Inui, K. (2014). An Example-Based Approach to Difficult Pronoun Resolution. In Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing (pp. 358-367).

[66] Turing, A. M. (1950). Computing machinery and intelligence. Mind, 59(236), 433-460.

[67] Ackerman, E. (2014). A better test than turing [News]. IEEE Spectrum, 51(10), 20-21.

[68] Banerjee, S., and Pedersen, T. (2003, August). Extended gloss overlaps as a measure of semantic relatedness. In Ijcai (Vol. 3, pp. 805-810).

[69] Budanitsky, A., and Hirst, G. (2001, June). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In Workshop on WordNet and Other Lexical Resources (Vol. 2, pp. 2-2).

[70] Hassan, S. (2011). Measuring semantic relatedness using salient encyclopedic concepts. University of North Texas.

[71] Islam, A., and Inkpen, D. (2009). Semantic similarity of short texts. Recent Advances in Natural Language Processing V, 309, 227-236.

[72] Jiang, J. J., and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008. .

[73] Leacock, C., Miller, G. A., and Chodorow, M. (1998). Using corpus statistics and WordNet relations for sense identification. Computational Linguistics, 24(1), 147-165.

[74] Lin, D. (1998, July). An information-theoretic definition of similarity. In ICML (Vol. 98, pp. 296-304).

[75] Mihalcea, R., Corley, C., and Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In AAAI (Vol. 6, pp. 775-780).

[76] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. International journal of lexicography, 3(4), 235-244.

[77] Patwardhan, S., Banerjee, S., and Pedersen, T. (2003, February). Using measures of semantic relatedness for word sense disambiguation. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 241-257). Springer Berlin Heidelberg.

[78] Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007.

[79] Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130-137.