A System for Building Corpus Annotated With Semantic Roles

Sanaz Rahimi Rastgar
Niloufar Razavi

MASTER THESIS 2013
INFORMATICS
A System for Building Corpus Annotated With Semantic Roles

Sanaz Rahimi Rastgar
Niloufar Razavi

Detta examensarbete är utfört vid Tekniska Högskolan i Jönköping inom ämnesområdet informatik. Arbetet är ett led i masterutbildningen med inriktning informationsteknik och management. Författarna svarar själva för framförda åsikter, slutsatser och resultat.

Handledare: He Tan
Examinator: Vladimir Tarasov

Omfattning: 30 hp (D-nivå)

Datum: 8 February 2013

Arkiveringsnummer:
Abstract

Semantic role labelling (SRL) is a natural language processing (NLP) technique that maps sentences to semantic representations. This can be used in different NLP tasks. The goal of this master thesis is to investigate how to support the novel method proposed by He Tan [1] for building corpus annotated with semantic roles. The mentioned goal provides the context for developing a general framework of the work and as a result implementing a supporting system based on the framework. Implementation is followed using Java. Defined features of the system reflect the usage of frame semantics in understanding and explaining the meaning of lexical items [2]. This prototype system has been processed by the biomedical corpus as a dataset for the evaluation. Our supporting environment has the ability to create frames with all related associations through XML, updating frames and related information including definition, elements and example sentences and at last annotating the example sentences of the frame. The output of annotation is a semi structure schema where tokens of a sentence are labelled. We evaluated our system by means of two surveys. The evaluation results showed that our framework and system have fulfilled the expectations of users and has satisfied them in a good scale. Also feedbacks from users have defined new areas of improvement regarding this supporting environment.
Acknowledgements

We would like to thank our supervisor, Dr. He Tan who supported perfectly during this master thesis work with her wise advices and guidance and also our examiner, Dr. Vladimir Tarasov for his useful advices and discussions on our thesis work.

We would like to kindly thank our families and friends whose moral supports encouraged us to successfully deliver this thesis.

Niloufar Razavi
Sanaz Rahimi Rastgar
Key words

Corpus Construction, Semantic Role Labelling, Semantic Roles, System Development, Frame Semantics
# Contents

1 Introduction ................................................................. 1  
1.1 BACKGROUND ............................................................... 1  
1.2 PURPOSE/OBJECTIVES ...................................................... 1  
1.3 LIMITATIONS .............................................................. 2  
1.4 THESIS OUTLINE .......................................................... 2  

2 Theoretical Background .................................................. 3  
2.1 NLP AND TEXT MINING APPLICATIONS .................................. 3  
2.2 SEMANTIC ROLE LABELLING .............................................. 4  
  2.2.1 Semantic Roles .......................................................... 5  
2.3 CORPUS ANNOTATED WITH SEMANTIC ROLES ....................... 6  
  2.3.1 FrameNet ............................................................... 6  
  2.3.2 PropBank ............................................................... 13  
  2.3.3 Semantic Role Labeling for Biomedical Domain ................ 15  
2.4 A NOVEL METHOD FOR CORPUS CONSTRUCTION ................. 17  
2.5 RELATED WORK .......................................................... 17  

3 Research Methods ......................................................... 19  
3.1 AWARENESS OF THE PROBLEM: LITERATURE REVIEW ........... 21  
3.2 SUGGESTION ............................................................. 23  
3.3 DEVELOPMENT: XP METHODOLOGY .................................... 24  
3.4 EVALUATION: DATA COLLECTION AND SURVEY ................... 27  
3.5 CONCLUSION PHASE ....................................................... 29  

4 Framework and System .................................................. 31  
4.1 FRAMEWORK .............................................................. 31  
  4.1.1 Framing ................................................................. 33  
  4.1.2 Annotation .............................................................. 34  
4.2 METHOD OF SYSTEM IMPLEMENTATION ................................ 34  
  4.2.1 User Requirements: User Stories .................................. 34  
  4.2.2 Software Development Environment ................................ 36  
  4.2.3 Interface Design ...................................................... 37  
  4.2.4 Java Class Design .................................................... 38  
  4.2.5 The Corpus Database ................................................ 39  
  4.2.6 System Requirements ................................................ 41  

5 Results and Discussion .................................................. 43  
5.1 THEORETICAL RESULTS .................................................. 43  
5.2 PRACTICAL RESULTS ..................................................... 44  
  5.2.1 Implementation Results .............................................. 44  
  5.2.2 Evaluation Result and Discussion .................................. 46  

6 Conclusion and Future Work ............................................. 49  
7 References ........................................................................ 51  
8 Appendix ........................................................................... 57
List of Figures

FIGURE 2-1: FRAMENET FRAME EXAMPLE ......................................................... 7

FIGURE 2-2: ANNOTATION LAYERS ........................................................................ 9

FIGURE 2-3: PROPBANK FRAME FILE EXAMPLE .............................................. 13

FIGURE 2-4: EXAMPLE OF FRAME DEFINITION .............................................. 14

FIGURE 2-5: AVAILABLE ARGUMENTS FOR AN EXAMPLE FRAME .................. 15

FIGURE 3-1: EVALUATION MODEL ..................................................................... 28

FIGURE 4-1: GENERAL FRAMEWORK OF SYSTEM ............................................ 31

FIGURE 4-2: "FRAMING" PROCESS OVERVIEW .............................................. 32

FIGURE 4-3: "ANNOTATING" PROCESS OVERVIEW ......................................... 33

FIGURE 4-4: COMPARISON BETWEEN DIFFERENT XML SCHEMAS ............. 39

FIGURE 4-5: XML FILE CONTAINING DATA THROUGH FRAME SEMANTICS ........ 40

FIGURE 4-6: OUTPUT FILE CONTAINING ANNOTATED TOKENS ..................... 41

FIGURE 5-1: ADDING NEW FRAME AND ITS DEFINITION ............................... 45

FIGURE 5-2: EVALUATION RESULT REGARDING EASINESS OF SYSTEM ............. 47

FIGURE 5-3: EVALUATION RESULT REGARDING EASINESS OF SYSTEM .......... 47
FIGURE 8-1: ADDING FRAME ELEMENTS AND DEFINITION TO A FRAME ......................................................................................................................... 57

FIGURE 8-2: EDITING A FRAME AND ITS RELATED OPTIONS .......... 57

FIGURE 8-3: SELECTION SENTENCE FOR ANNOTATION PURPOSE..... 58

FIGURE 8-4: CONFIRMING THE SELECTED SENTENCE TO BE ANNOTATED ........................................................................................................... 58

FIGURE 8-5: TOKENISATION OF SENTENCE ............................................. 59

FIGURE 8-6: SETTING ROLES TO TOKENS ............................................. 59

FIGURE 8-7: SAVING THE RESULT THROUGH FILLED TABLE ............ 60

FIGURE 8-8: ANNOTATE USING POS TAGGER ........................................ 60

FIGURE 8-9: DIVIDING TOKENS ................................................................. 61

FIGURE 8-10: SURVEY "EASINESS" ........................................................ 63

FIGURE 8-11: SURVEY "SPEED" ............................................................... 64
List of Abbreviations

**ADVP:** Adverbial Phrase
**AI:** Artificial Intelligence
**DTD:** Document Type Definition
**DSD:** Document Structure Description
**DSR:** Design Science Research
**FE:** Frame Element
**GF:** Grammatical Function
**GUI:** Graphic User Interface
**IDE:** Integrated Development Environment
**IE:** Information Extraction
**IR:** Information Retrieval
**IS:** Information System
**LU:** Lexical Unit
**NLP:** Natural Language Processing
**NP:** Noun Phrase
**PAS:** Predicate Argument Structure
**POS:** Part-Of-Speech
**PP:** Propositional Phrase
**PT:** Phrase Type
**SOX:** Schema for Object-Oriented XML
**SRL:** Semantic Role Labelling
**ST:** Semantic Type
**TM:** Text Mining
**XML:** Extensible Markup Language
**XDR:** XML-Data Reduced
**XP:** Extreme Programming
Introduction

This section delivers general understanding of our work by means of discussing related concepts in background part. Also the research questions are exposed to discussion which gives direction to reaching the research’s objective.

1.1 Background

The study of Semantic Role Labelling (SRL) is an important notion in the field of text mining, information extraction (IE) and Natural Language Processing (NLP) as it helps interpreting sentences on semantic level [3]. SRL deals with identifying the semantic roles or relationships in a sentence structure within a semantic frame [4]. This is informally knows as assigning “who” did something and “what” was the thing did, to “whom, when, where, why, how, etc.” [3]. During the past years, projects such as PASBio, BioProp and BioFrameNet have made lots of efforts in the biomedical domain to apply SRL. However, the development of SRL systems for biomedical domain faced problem by the lack of large corpora for such a domain. The problems appeared due to difficulties in defining frames with their associated roles, grouping example sentences to each semantic frame as well as collecting them from databases [1].

Recently a method is proposed by He Tan [1] for building corpus which is labelled by semantic role labelling for the biomedicine’s domain. The method makes use of domain knowledge provided by ontology. By this method, a corpus has been built which is related to biological transport events. In this master thesis we have reviewed similar concepts and systems to discover how to support this method of semi-automatic labelling. As an important step towards fulfilling the objective, formulating correct research questions has a vital role. We have mentioned the research question as followed in next section.

1.2 Purpose/Objectives

A method of building corpus with frame semantics annotations using domain knowledge provided by ontologies was developed by He Tan [1]. By using the method, they have successfully built a corpus of biological transport events that are based on the domain knowledge provided by GO biomedical process ontology [1].

The purpose of this thesis work is formulated in three research questions as follows:
Introduction

1. How to support the method of building corpus annotated with semantic roles using ontological knowledge.

2. What general framework is needed regarding support of the novel method?

3. How to implement a semi-automatic system based on the general framework to support semi-automatic this kind of corpus construction.

1.3 Limitations
According to fulfilling the objectives of system which was explained previously by means of three research questions, we did not find any limitation. As long as the system delivers the expected goals, no limitations are exposed to discuss. Currently the system is based on data in biomedical domain but it can be used in other fields as well.

1.4 Thesis outline
This document is structured as six chapters:

- Chapter 1 which has covered the outline of thesis is basically introduction part that presents semantic role labelling, background and objectives of work.
- Chapter 2 gives the definitions for main concepts used in the system, the previous related approaches.
- Chapter 3 describes the research method followed for reaching the thesis goals.
- Chapter 4 deals with introducing framework and system overview as well as explaining the method used for system implementation.
- Chapter 5 presents the results achieved during the thesis work.
- In chapter 6, the results and findings are consolidated in conclusions as well as some ideas for further research are presented.
2 Theoretical Background

This chapter will include the basic knowledge regarding the development of SRL systems, how they work and why they are important in text mining applications, so that a reader of this research could understand basics of the development process which is related to objective of our thesis.

2.1 NLP and Text Mining Applications

Text mining is the process of discovery and extracting interesting information from unstructured text. This involves everything from information retrieval, lexical analysis, to information extraction. The main objective of these applications is turning text into data for analysis by the means of NLP and analytical methods [5].

NLP methods try to extract a fuller meaning representation from text. One task on the semantic level can be described as finding out who did what to whom, where, when, how and why. Regarding this explanation, SRL can be seen as a task of NLP which we have described more in details in section 2.2. In fact, NLP makes possible the use of linguistic concepts, for instance part-of-speech (POS), (such as noun, verb, adjective, etc.) and grammatical structure [6]. In other words, different techniques have been developed by NLP which have typically got inspiration from linguistic concepts. An example is parsing a text syntactically by using formal grammar information or lexicon information. Next step is interpretation of resulting information in a semantically way [7]. Working with linguistics concepts and grammatical structure, considerably causes dealing with anaphora and ambiguities where anaphora is about “what previous noun does a pronoun or other back-referring phrase correspond to” and ambiguities is about “both of words and of grammatical structure, such as being modified by a given word or prepositional phrase” [6]. For this matter it is important to get benefit of several knowledge representations such as:

- Lexical unit (lexicon of words and their meaning)
- Grammatical properties
- A set of grammar rules
- Thesaurus of synonyms and abbreviations
- Other resources such as ontology of entities and actions [6].
Several tasks approached by using text mining techniques, mainly split into two groups. Some of them such as information retrieval, text categorization, and document clustering operate on the document level, while the others like document summarization, IE, and question answering operate on the sentence level [5]. In fact both of the mentioned groups are affected by the problem of “data sparsity” for modelling the language accurately, where the most emphasize is on the latter group [5, 8]. The term of data sparsity is used for giving explanation to the phenomenon in case of not considering enough data in a corpus to model language accurately [8]. Lack of data causes problems in observing the true distribution and pattern of language [8]. The nature of the text mining task as well as the domain of interest, are other issues that need to be considered.

Text mining technology is broadly applied for various research needs. It has also lead to creation of different applications like biomedical applications or even marketing applications. Text mining from biomedical text has grown to be one of the main issues in bioinformatics field and NLP methods have been used to increase the potential of text mining from biological text [9].

2.2 Semantic Role Labelling

Automatic semantic role labelling is the task in NLP that maps free text sentences to the semantic representations. The task simply is to identify all parts of a sentence and label them with a semantic role for a given predicate [10]. Therefore the input of the SRL system is a sentence and a predicate (or target) in that sentence, the output is labelled sentence with semantic roles. In order to approach SRL, independently of one’s background, overall understanding of the theory of semantic roles should be examined.

SRL is sometimes known as shallow semantic parsing which is consisted of recognition of the semantic arguments associated with the predicate or verb of a sentence. It also includes their classification into their specific roles [11]. We can clarify the concept of semantic role labelling with an example:

Assuming the sentence “Anna sold the book to Marcus”, steps towards making the meaning of sentence clear are:

- Recognizing the verb “to sell” as representing the predicate
- Recognizing “Anna” as representing the seller (agent)
- Recognizing “the book” as representing the goods (theme)
- Recognizing “Marcus” as representing the recipient
As it is shown, SRL is a shallow semantic processing task that has become increasingly popular in NLP community over the last few years. The task is to identify all parts of a sentence that represent arguments for a given predicate and subsequently label each argument with a semantic role. Roughly speaking, SRL can be thought of as the task of finding the words that answer simple questions of the form who did what to whom when and where? The input to the SRL system is a single sentence and a predicate in that sentence. The output is the same sentence, but with labelled semantic roles.

The most important computational lexicons were created by FrameNet project and PropBank. A vast amount of predicates and the corresponding roles of those predicates were defined systematically by the lexicon. The first automatic semantic role labelling system was developed based on FrameNet by Daniel Gildea and Daniel Jurafsky [12].

### 2.2.1 Semantic Roles

The relationship which a syntactic constituent has with a predicate is called semantic roles. Agent, patient and instrument create typical semantic arguments [13]. Answering “WH” questions such as "Who", "When", "What", "Where", "Why" in Information Extraction, Question Answering and Summarization, needs recognition and labelling semantic arguments. In general labelling semantic arguments play a key role in the NLP tasks which are related to some kind of semantic interpretation. There are different schemes for specifying semantic roles where the commonly used schemes out of them, are the PropBank annotation scheme and FrameNet [14]. The PropBank is based on Penn TreeBank and its corpus added semantic role annotations which are created manually to the Penn TreeBank corpus of Wall Street Journal texts. PropBank has been used by many automatic semantic role labelling systems as a training dataset which usage helps understanding how to annotate new sentences automatically [11, 15]. The FrameNet project has the key concept of annotation using frame semantic which supports creating a lexical resource [16].

Semantic roles, also known as thematic roles, are one of the oldest construction classes in linguistic theory. Semantic roles are used to indicate the role played by each entity in an event apart from linguistic encoding of that event [11]. For example if someone named John hits someone named Bill, the John is the agent and Bill is the patient of the hitting event. Agent and patient are the semantic roles in following sentences:

John hit Bill.

Bill was hit by John.
In both of above sentences, the semantic role of Bill is patient and John has the semantic role of agent. Although there is no consensus on a list of semantic roles, some basic semantic roles like agent, patient, theme, location, source and goal are followed by all.

Correctly identifying semantic roles of a sentence is a crucial part in sentence-level text mining applications. Following paraphrases show that for a single predicate, semantic arguments can have multiple syntactic understandings:

John will meet with Mary.
John will meet Mary.
John and Mary will meet.

The theoretical status of semantic roles in linguistic theory is still unsolved. There is an uncertainty about whether semantic roles should be observed as syntactic or semantic entities. However the most common appreciative is that semantic roles are conceptual elements as a way of classifying the arguments of a sentence [17].

2.3 Corpus Annotated with Semantic Roles

There are different ways of annotation a corpus with semantic roles. Two related work are discussed here to demonstrate how these projects process documents by means of SRL. These literature reviews provide us the knowledge regarding how the text is processed. They also provide us a perspective to investigate the way for supporting the mentioned method.

2.3.1 FrameNet

FrameNet is a lexical database based on the Frame Semantics theory that labels words in a sentence. A word is stored by its meaning as a pair titled Lexical unit (LU). Each predicate (target word) in a sentence and its arguments is associated to a frame. The basic unit of this framework is the frame, initially defined as a type of an event and its contributors called frame elements (FEs). An example that shows a sentence annotated with FrameNet is provided to explain concepts [17]:

[Cook Matilde] fried [Food the catfish] [Heating_instrument in a heavy iron skillet].

In this example, the target word “fried” evokes the frame “Apply_heat”. “Apply_heat” describes a situation involving a “Cook”, some “Food”, and a “Heating_instrument”. These are called frame elements. Frame evoking words like bake, boil, steam, fry, etc. are LUs in the “Apply_heat” frame that also can be a target word of annotated sentence.
For representing a schematic view of semantic knowledge better, another example of frame in FrameNet can be shown in figure 2-1. In this example, the GIVING frame relates the frame elements of verb *Give* to the *Donor, Recipient and Theme* semantic roles. Other verbs that evoke the GIVING frame are represented in LU.

<table>
<thead>
<tr>
<th>Frame: Giving</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lexical units:</strong> give.v, give_out.v, hand in.v, hand.v, hand_out.v, hand_over.v, pass.v.</td>
</tr>
<tr>
<td><strong>Frame elements:</strong></td>
</tr>
<tr>
<td>Donor: The person that begins in possession of the Theme and causes it to be in the possession of the Recipient.</td>
</tr>
<tr>
<td>Recipient: The entity that ends up in possession of the Recipient.</td>
</tr>
<tr>
<td>Theme: The object that changes ownership</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
</tr>
<tr>
<td>[300 euro]_theme was given [to John]_recipient [by his mother]_donor</td>
</tr>
</tbody>
</table>

Figure 2-1: FrameNet Frame Example [18]

The FrameNet database is different from other dictionaries and thesauri with its exclusive characteristics [17]:

The main corpus is 100 million words British Natural Corpus (BNC). Analysis of the English lexicon proceeds frame by frame rather than word by word, what is done in traditional dictionaries. It provides a multiple annotated examples of each lexical unit which illustrates all of the combinations of that lexical unit. Each lexical unit is related to a semantic frame and also to other words which activate that frame.

FrameNet provides a set of relations between frames including: Inheritance, Using, Sub frame, Perspective on, and etc. However the FrameNet database cannot be used as ontology of things, since there are many nouns and artefacts which are not annotated. Daily work is made up of define a frame and its FEs, LUs (list of words evoke the frame), extract example sentences relate to the frame and annotate them. Annotation part is done by realization of FEs, phrase type (PT), and grammatical function (GF). FrameNet comprises three main parts [19]:

- A Lexical unit database containing pair of word and related frame (used to meaning of a word).
- A frame database entailing a set of frames, associated frame elements, and relations between frames.
- An example sentence database including a collection of lexical indication for frames used as a training set for labelling.
Frame development Process begins by searching corpus attestation of a group of words that seems to have some semantic overlap. Later these attestations are divided into groups to make frames in the reasonable point by target words, lexical units, and frame elements. This idea is durable to assess since there were some exceptions need to be managed separately. Following are some criteria used to form the groups of frames [17]:

- All LUs in a frame should have the same types of frame elements with the same set of transitions.
- The same frame elements must outlined across all lexical units of a related frame.
- The same interrelations between frame elements should exist for all the LUs in the frame.
- The basic denotation of target words should be similar in a frame.
- Specifications of the frame evoking words give to all frame elements of a frame should be similar.

The routine work of FrameNet consists mainly of annotating sentences chosen from a corpus as examples of a particular lexical unit [17]. Initially, the emphasis of annotation was on what was most relevant to lexical descriptions, namely the core and peripheral frame elements of target words. Its goal is to annotate words or phrases in a sentence that have relation in grammatical construction with the target word.

For each target word, there is a set of annotation layers for the FEs, phrase types, grammatical functions, etc. Each such set is represented by an entry in the Annotation table. In addition to the FE, GF, and PT layers, annotators also add labels on other layers, all of which are represented similarly. Certain syntactic information is represented by adding labels on the part-of-speech-specific layer [17]. In choosing the phrase types and grammatical functions; the major criterion was whether or not a particular label might figure into a description of the grammatical requirements of one of the target words.

The annotation start with labelling parts of the example sentences with tags indicating relevant syntactic and semantic properties. Figure 2.2 shows annotation layers of the following example sentence in the “Perception-passive” frame: “Helmut saw a tall, black figure against the shining snow.”

A component of sentence may express a particular frame element such as Helmlut states the FE “Perceiver-passive” a tall, black figure, the FE “Phenomenon”, and against the shining snow, the FE “Ground”. Next layer of annotation is to specify phrase types of each of these constituents. Further grammatical function regarding target word (“see” in the example) is described. These three layers are independent called FE, PT and GF [17].
Below, the detailed concepts of semantic annotation of natural language texts used in FrameNet project are described:

### 2.3.1.1 Frame Semantics

Frame semantics starts with the assumption that in order to understand the meanings of the words in a language, we must first have knowledge of the background, motivation for their existence in the language and their use in discourse [20]. The knowledge can be provided by the conceptual structures, or semantic frames.

A frame semantic view would relate each of the relevant words to the background frame. In a technical language, it is easy to support the association of word to frame but in some lexical fields, for instance biomedical domain, semantic theory is not enough to find the relevance of terms to the frames.

According to the definitions above, the most important issue about frame semantics is remembering the task of frame semantics: understanding and explanation of lexical items meanings as well as grammatical constructions [2]. As an extension of Charles J. Fillmore’s case grammar [21], it relates linguistic semantics to encyclopaedic knowledge. In other words the assumption in frame semantics is formulated in this sentence: understanding the meanings of the words of a language needs having knowledge. This knowledge refers to the conceptual structures, or semantic frames, which underlie their usage. For example, while talking about “sell”, one would be able to realize the meaning of word "sell" if only he has a knowledge about the state of commercial transfer, which also includes other features such as a seller, a buyer, goods, money, the relation between the money and the goods, the relations between the seller and the goods and the money, the relation between the buyer and the goods and the money and so on [22].

According to the frame semantics idea presented by Charles Fillmore’s [21] frames work as a type of cognitive structuring device which the background knowledge and motivation for the existence of words in a language is provided by them as well as understanding their usage in discourse [2, 23].
The vast variation of approaches in systematic description of natural language meanings is also discussed by the term frame semantics. There is something common among these approaches which can be followed due to Charles Fillmore’s saying: this saying states that meanings own internal structure which are moderately determined to a background frame or a scene. This common feature does not sufficiently make a distinction between frame semantics and other frameworks of semantic description [24].

Two historical roots of frame semantics are available. First root centres on linguistic syntax and semantics which is mainly about Fillmore’s case grammar, the other one is in the field of Artificial Intelligence (AI) and centres around the concept of frame introduced by Minsky [25].

To become in details, first discussion refers that within a case grammar, case frame was used for characterizing a small abstract scene with the goal of identifying the participants of the scene and as a consequence identifying the arguments of predicates and sentences which are describing the scene. It is assumed that in order to understand the sentence, the language user has mental access to such schematized scenes.

Although discussion about second history root is difficult, details express that it is about concept of frame-based systems of knowledge representations in AI. This root of frame semantics is known as highly structured approach to knowledge representation, which goal is arranging the collected information about specific objects and events into a taxonomic hierarchy familiar to biological taxonomies [24].

2.3.1.2 Frame
A semantic frame describes an event, a situation or object, together with the participants (called frame elements (FE)) involved in it. A word evokes the frame when its sense is based on the frame. The relation between frames include is-a, using and subframe [23]. A collection of facts forms a frame which this collection identifies "characteristic features, attributes, and functions of a denotatum, and its characteristic interactions with things necessarily or typically associated with it." [26]. It can also be defined as a coherent structure of related concepts which without knowledge of all the related concepts, understanding is not possible.

Words do not only focus on individual concepts, they also specify an assured perspective from which the frame is viewed. For example "sell" defines the situation from the perspective of the seller and "buy" from the perspective of the buyer.

2.3.1.3 Frame Elements
Frame elements are the participants, props and roles of a frame including agents and objects [27]. They are also well-defined for their syntactic dependents role of a predicating word. Each FE is linked relevant to a single frame.
FEs are divided into core, peripheral and extra-thematic, according to how central they are to a frame. A core FE is theoretically necessary to a frame due to the situation described in a frame. A peripheral FE is usually repeated in different frames and marks such notions as Time, Place, Means, etc. and therefore does not characterize individually a frame. Extra-thematic FE type is somehow different from the peripheral type and introduces an extra state or event. For this reason, these FEs don’t theoretically belong to the frame they seem to be and have a somewhat independent status. These types of frame elements have also the ability to evoke a larger frame embedding the reported situation [4].

As we described frame semantics by the example of “sell”, the frame and frame elements are recognisable in which the frame is commercial transaction and frame elements are Buyer, Seller, Goods, and Money. This example is most often cited example of Fillmore’s about frame semantics. In this frame, Lexical units belonging to this frame are verbs such as buy, sell, spend, or charge, nouns such as price, goods, or money, and adjectives such as cheap and expensive. While all of these lexical units belong to the same semantic frame (the commercial transaction frame), a specific choice of a lexical unit reveals a particular perspective from which the commercial transaction frame is viewed [28].

2.3.1.4 Lexical Unit
A lexical unit is a pair composite that represents a word and a meaning. Lexical unit is different from word and is associated to a semantic frame [17]. For example if the word bake (which has the word forms: bake, bakes, baked, and baking) is linked to three different frames: Apply_heat, Cooking_creation and Absorb_heat, Multiple expressions of the word bake in each of the above frames, construct three different lexical units (and not the word forms). In some lexicographic work, annotation is done to a lexical unit in the sentence, which is a target word.

2.3.1.5 Target Words
Given an example sentence, the word with semantic and syntactic properties of interest is called the Target Word or simply the target [17]. A target word can be in any of the major lexical categories: a noun, verb, adjective, adverb or preposition. In annotation process, a sentence from different texts of a corpus is extracted by a predetermined target word and then frames are evoked by target words. In order to annotate a collection of example sentences for a certain target word, it is necessary for the annotators to understand the frame linked with that word by getting access to the provided frame definition.

2.3.1.6 Example Sentence
The mainly work of FrameNet involves annotating example sentences extracted from a corpus for a specific lexical unit. A software is used to choose example sentences for a LU. The sentences are presented to the annotators and grouped to patterns. The reason behind alignment example sentences is to annotate
easier and make sure to annotate a few examples of each different pattern. Since there is a set of annotation layers for each target of an example sentence, each such set is represented in the annotation file with linking a sentence and LU [29].

### 2.3.1.7 Phrase Type and Grammatical Function

The syntactic Meta language used in the annotation process called phrase Type (PT). In order to annotate words in a sentence this notion is used to show lexical descriptions of terms in respect to the target word. Identifying phrase type is important to distinct each frame element. Phrase types are assigned manually by the annotators during the annotation process. What follows is a list of phrase types that are used in the system, complemented by some examples [17].

- **Noun Phrase (NP):** Standard Noun Phrase that can fill core argument slots.
  
  [My neighbour] is a lot like my father.
  [John] said so, too.
  [You] want more ice-cream?

- **Prepositional Phrases (PP):** With NP object Prepositional Phrases are assigned.
  
  Scrape it back [into the microwave bowl].

- **Adjective Phrase Types (AJP):** It is used for relational modification of adjectives.
  
  Philip has [bright green] eyes.
  The light turned [red].

- **Adverb Phrase (AVP):** used for adverbs.
  
  All items at [greatly] reduced prices!

- **Verb Phrases (VP):** Verb phrase can be a main verb or an auxiliary.
  
  This book [really stinks].
  I didn’t expect you to [eat your sandwich so quickly].

In annotating example sentences, each constituent is tagged with a frame element related to a target word. These constituents that are tagged with frame elements are assigned grammatical function in respect to that target word. Grammatical function (GF) defines in which ways the constituents fulfil grammatical requirements related target word. Examples of the grammatical functions used in the system are [17]:

---

*Theoretical Background*
2.3.2 PropBank

PropBank or proposition Bank project precedes a practical approach to semantic representation. It is built by the aim of adding a layer of semantic annotation consists of predicate-argument information or semantic role labels, to the Penn Treebank [30, 31]. The idea of PropBank creation was mainly serving as training data for machine learning-based semantic role labelling systems. According to this idea, it is necessary that all arguments of verbs be syntactic constituents and distinguish of various meanings of a word is only possible if the differences bear on the arguments [32].

The focus of proposition bank is on the argument structure of verbs which has made it known as a verb-oriented resource [30]. It provides a complete corpus annotated with semantic roles which the roles are seen as arguments and adjuncts [30]. In other words, the main option which has made PropBank different from FrameNet is that annotation is based on verb-specific roles [31]. According to the fact that PropBank’s task is annotating all verbs in a corpus, annotation of events or states of affairs which is termed by usage of nouns, is not committed by PropBank. Annotation done by PropBank almost stays closely to the syntactic level [33].

The lexicon of PropBank has defined frame files for all the verbs which every verb is the owner of a unique frame file. A frame file is consisted of specific role sets for every word-sense of the verb. Verbs are known as predicates in PropBank, so each predicate refer to the verb in PropBank. The grouping of predicate and related arguments is called a proposition [31]. An example of a frame file for the verb *Give* is shown below:

```
<table>
<thead>
<tr>
<th>Roleset give.01 &quot;transfer&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roles:</td>
</tr>
<tr>
<td>Arg0:  Giver</td>
</tr>
<tr>
<td>Arg1:  Thing given</td>
</tr>
<tr>
<td>Arg2:  entity given to</td>
</tr>
</tbody>
</table>

| Example:                    |
| [The executives]_{arg0} gave [the chef]_{arg2} [a standing ovation]_{arg1} |
```

Figure 2-3: PropBank Frame File Example [18]
Using PropBank makes the possibility of practically specifying frequency of syntactic variations, the troubles they bring up for natural language understanding, and the policies to which they may be tractable [30].

Due to the numbering of core arguments from 0 to 5 which is listed as ARG0, ARG1, ARG2, ARG3, ARG4 and ARG5, they are called numbered arguments. These arguments are specific for each verb sense. Beside the numbered arguments specific to the verbs, they can be assigned to a set of general arguments also. These general arguments are called ARGMs (verb modifiers). ARGMs can be compared to non-core elements in FrameNet due to the fact that they are not verb specific.

As a middle ground among many theoretical theories, numbered arguments theory was selected and used in PropBank. The reason of choosing this theory was the possibility in mapping consistently of numbered arguments to any theory of argument structure [30]. PropBank gets benefit of Levin classes of verbs in order to label verbs consistently. To understand the work process it is better to take a look again at Fillmore’s theory.

Fillmore states that a relation exists between theta roles (deep cases) and grammatical functions, an example can be the role of the subject of a transitive non-passive verb which generally corresponds to the agent role, the direct object to the patient role: [Anna Subject, ARG0, Agent] eats [the chocolate cake direct object, ARG1, Patient]

It should be considered that the grammatical function of the patient role can be modified if changes happen in the way verbal arguments are grammatically expressed. These changes are called diathesis alternations:

Middle Alternation: [The chocolate cake Subject, ARG1, Patient] smells perfect.

In the first example the direct object plays the role of ARG1, while in second example the ARG1 role of smell is expressed by the subject.

In Levin’s verb classification [34], verbs which are sharing the same diathesis alternations, share the same argument structure. According to the efforts done in PropBank, it was ensured that verbs belonging to the same class are given consistent role labels. The verb “wonder” can be taken into account as an example of frame definition in PropBank. The frame definition is shown in the figure 2-4:

![Figure 2-4: Example of frame definition [16]](image-url)

Roleset wonder.01 "think about, ponder"
ARG0: thinker
ARG1: thought
The verb wonder takes two core arguments: arg0 and arg1. Additionally it can take any number of ARGMs like any other verbs. Figure 2-5 shows a summary of available ARGMs.

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgM-LOC</td>
<td>Location</td>
</tr>
<tr>
<td>ArgM-EXT</td>
<td>Extent</td>
</tr>
<tr>
<td>ArgM-DIS</td>
<td>Discourse connectives</td>
</tr>
<tr>
<td>ArgM-ADV</td>
<td>General purpose</td>
</tr>
<tr>
<td>ArgM-NEG</td>
<td>Negation mark</td>
</tr>
<tr>
<td>ArgM-MOD</td>
<td>Modal verb</td>
</tr>
<tr>
<td>ArgM-CAU</td>
<td>Cause</td>
</tr>
<tr>
<td>ArgM-TMP</td>
<td>Time</td>
</tr>
<tr>
<td>ArgM-PNC</td>
<td>Purpose</td>
</tr>
<tr>
<td>ArgM-MNR</td>
<td>Manner</td>
</tr>
<tr>
<td>ArgM-DIR</td>
<td>Direction</td>
</tr>
</tbody>
</table>

Figure 2-5: Available Arguments for an example frame [16]

The example below is an example sentence taken from PropBank corpus which shows the annotation process of a complete proposition:

[They ARG1] are [n’t ARGM–NEG] [accepted REL] [everywhere ARGM–LOC], [however ARGM–DIS].

Development process of PropBank is consisted of two parts: framing and annotation:

- Framing: First step in the framing process is to examine a sample of sentences. These sentences are from the corpus which they include verbs in their structure. Next step is grouping the instances into one or more major senses where later each of them converts to a single frameset [30].
- Annotation: First step in the annotation process was running a rule-based argument tagger on the corpus. Second step is correcting the tagger’s output manually. PropBank corpus annotation is a two-pass process where each verb is annotated by two annotators tracked by an adjudication phase to resolve alterations between the two initial passes [31].

2.3.3 Semantic Role Labeling for Biomedical Domain

The ability to accurately identify the meanings of terms is an important step in automatic text processing. It is necessary for applications such as information extraction and text mining which are important in the biomedical domain.
Data in Biomedical area text significantly is different from FrameNet and PropBank. Like text in other domains, biomedical documents contain a range of terms with more than one possible meaning. These ambiguities form a significant obstacle to the processing of biomedical texts. Currently there are some approaches to resolving this problem but no large corpus for such a domain exists.

Biology improvements have led to a great growth in the amount of biomedical literature. Thus, automatic information retrieval and information extraction methods become more and more important to help researchers to get to know of the latest developments in this field. Current IR is still mostly limited to keyword search especially when it is needed to infer the relationship between two entities in a text. Understanding how words are related in a sentence is an important factor to improve both the quality of IE systems and the ability to search more complex queries by IR systems.

There are some difficulties in adapting semantic role labelling technologies to new domains such as biomedical domain. These problems can be divided into two main categories: differences in text style and differences in predicates. The CoNLL 2005 shared task [35] explored semantic role labelling systems that were trained on the Wall Street Journal and were established on the Brown corpus. After comparing results on the Wall Street Journal data, they found that "all systems experienced a severe drop in performance ". Reason of the drop was mainly poorer performance of sub-components like part-of-speech taggers and syntactic parsers. Researchers have found a similar performance drop was where training semantic role labelling corpus on nominal predicates.

Pradhan et. al. [36] reached only an F-measure of 63.9 when evaluating their models on nominal predicates from FrameNet and some manually annotated nominalizations from TreeBank. Jiang and Ng [37] achieved better results on the NomBank [38] corpus, but their F-measure was still only 72.7 that were more than 10 points below normal performance for verbs. Therefore, these research efforts suggest that adapting semantic role labelling to biomedical domain involves some remarkable challenges.

One of the SRL systems that targeted in the biomedical text is BIOSMILE [39]. The BIOSMILE system was trained on BioProp corpus [40], a biomedical proposition bank semi automatically annotated in the style of PropBank. However BioProp, similar to other biomedical corpora with predicate argument structures reflected only verbs, such as Kogan and colleagues corpora [41]. It annotated 30 biomedical verbs in 500 abstracts. Our work significantly differs from BIOSMILE in corpus construction method. Both the data and algorithm that were used in this work is different. In BIOSMILE semantic roles are only allowed to match full syntactic units because BioProp followed PropBank style. We consider all data includes multi-word roles for handling nominal predicates describing Transport events. Because of these many differences in text comparing to other domains, we reconnoitred an alternative to the syntactic
constituent approach used by BIOSMILE. Consequently studying models allow us to evaluate methods that did not rely on syntactic parses.

2.4 A Novel Method for Corpus Construction
There are difficulties in construction a large corpus for domain based systems by frame semantic. To ease the task ontologies, as a semantic representation domain based knowledge, are used [1].

A method for building corpus that is labelled with semantic roles for the domain of biomedicine is introduced by He Tan. This method is based on the theory of frame semantics, and relies on domain knowledge provided by ontologies. By using the method, they have built a corpus for transport events strictly following the domain knowledge provided by GO biological process ontology.

Ontology is a shared and common understanding of some domain which can be defined as a conceptualization in order to support a specification, i.e., ontology defines entities and relationships among them. Therefore ontology can be used as a solution by describing all possible events and translating them into frame.

The successful corpus construction demonstrates that ontologies, as a formal representation of domain knowledge, can instruct us and ease all the tasks in building this kind of corpus [1]. Furthermore, ontological domain knowledge leads to well-defined semantics exposed on the corpus, which will be very valuable in text mining applications.

In this thesis, we aim to develop a supporting environment which has the components that support parsing and visualizing lexical properties of ontological terms, defining frame semantics description and annotation task for such a corpus construction method.

2.5 Related Work
In this section, we have discussed some differences and similarities of other projects to our project. FrameNet and PropBank are explained in section 2-3. Understanding features of these projects helped us to find out the challenges related to the work. We came to the decision at how we support the new method by improving the existing system used in FrameNet project. Below you can find explanation regarding this issue:

We have studied related works and used some common tools out of them. The similarity of PropBank, FrameNet and our project is visible in terms of goal which is presenting a semantic annotation layer for corpora. In fact the goal is the same, but achieving the goal is different according to the existed problems. As it was discussed in SRL systems, a problem is lack of large corpora in
biomedical domain because data in biomedical area texts is significantly different from FrameNet and PropBank. For example many words exist in biomedical domain which they never appear in general English and biomedical documents contain a range of general English terms with very specific meaning for the domain. These problems form a significant obstacle to the processing of biomedical texts using FrameNet and PropBank which developed for general English. Currently there are some approaches to resolving this problem but no large corpus for such a domain exists. This challenge is responded by a solution to consider all possible biomedical events. Our supporting system solves this problem by help of frame semantics which gives the opportunity to have all possible events available in the corpus by using the described method. This is done by means of defining new frames and its associations using domain knowledge provided by ontologies. Using frame semantics can be referred as a similarity of our project to FrameNet. Although we have different semantic senses from FrameNet, the work demonstrated here shows that FrameNet style in annotation can be integrated with information from ontologies to discover and define frames. Another challenge is that the method used in this supporting system, can tokenise and annotate all arguments as well. The advantage of our work can be considered as: supporting environment for ontology-based method. This means it has the possibility of extending with ontology as it supports ontological method.
Research Methods

3 Research Methods

In order to create a new knowledge, or make the knowledge about a subject deeper, choosing the appropriate research method is necessary. There are two main ways for research design: qualitative research and quantitative research. One of these research design types are chosen by researchers depending on the research problem which is going to be observed or the research question which is going to be answered. Different research methods are discussed in different resources, for example Williamson has described methods are such as action research, experimental research, etc. [42] and Ghauri & Gronhaug have described methods are such as exploratory research, descriptive research, casual research and case study [43]. Different steps of conducting this master thesis as a research are explained in this part.

In this master thesis we have generally used design science methodology to present a design science research. Besides, we got benefit of literature review method for theoretical contribution in this subject, and then the implementation method is used to implement a system related to the research’s subject.

Design science is a methodology in scientific researches which is mainly used in researches in the field of information systems (IS). This kind of researches focuses on development and performance of artifacts with a clear intention of improving the performance of artifacts [44].

Design science paradigm was used in many of the initial researches of IS which their focus was on systems development approaches and methods. Examples of these kinds of researches are discussion on the socio-technical approach (defined by Bostrom and Heinen [45] and Mumford [46], and the info-logical approach defined by Langefors [47], Sundgren [48], Lundeberg et al. [49]) [50]. Basically the design science paradigm is a problem-solving paradigm [51] which the foundations exist in engineering and the sciences of the artificial [52, 53].

Design process is consisting of some different phases which are defined by various DSR frameworks. This division of phases are done by specifying a set of milestones during the design process. These DSR frameworks usually launch repetitive approach including several phases of the design process. The example of this type can be referring to [54, 55, 56].

Different researches have different agreement on using variety of phases in a design science process. According to this fact, a common understanding about critical phases within a design science process helps this issue. Knowing these critical phases makes the possibility of choosing vital research activities for each phase. Related activities include inductive and deductive steps which are essential to build design principles regarding to a practical problem.
Deductive steps move towards developing more concrete design decisions. These steps include some activities routing to an instantiated artefact as well as methods routing to a comprehensive evaluation concept with the aim of permitting generalizations. Inductive steps concern about underlying design principles and theories [50].

As we said, the roots of design science paradigm is in science of artifacts which this feature has made the attention of researches towards the ‘design’ of artificial artifacts (i.e., IT artifacts) and generating new issues which are not yet exist. Using design science paradigm in a research is considerable because of the nature of science which is both known as a process (set of activities) of ‘creating something new’ and a product (i.e., the artifact that results out of this process) [57,58].

Characteristic of design science research can be described in these lists:

- The primary focus in a design science research project concentrates mostly on design research part (i.e. the creation of an IT artifact), opposite to the design science part (i.e. generating new knowledge).

- One of the issues involved in the design science research process is searching for a relevant problem, the design and construction of an IT artifact, and its ex ante and ex post evaluation.

- Searching for real-world problems and solving them practically is one of the most important goals in design science research.

- Design science research is a general research approach with a set of defining characteristics and can be used in combination with different research methods.

- Design science research is conducted most frequently within a positivistic epistemological perspective.

- The outcome of design science research (i.e., the problem solution) is mostly an individual or local solution and the results cannot be readily generalized to other settings [58].

We have used this methodology for providing this master thesis including several steps for making design and later implementing the defined method. The following steps cover the context of research such as knowing and stating the problem, suggestion for solving the problem and implementation of the suggested solutions. Last steps are about testing and evaluation the output.
Research Methods

These steps of design science research method (DSR) are illustrated as following [59, 60]:

1. Awareness of the problem
2. Suggestion
3. Development
4. Evaluation
5. Conclusion

We have followed these steps for using the methodology in our master thesis. Below we have described how we have translated the five steps in the process of our work. 3.1 to 3.5 explain how the structure has been fitted to the steps of design science methodology.

3.1 Awareness of the Problem: Literature Review

It is the first step in starting a research. Several different sources of information can lead to the awareness of the problem. These sources are such as: new development in industry or in a reference discipline [59, 60]. Recognizing the problem leads to a proposal which can be formal or informal for creating a new research [59, 60]. By reading different literatures related to the subject we have chosen, the awareness of problem was obtained for our research work. We stated the research questions for this matter.

There is a problem or research question which has been introduced in the beginning of a research and has been answered during that research. Result of the literature review can formulate the problem and become a motivation to the research work. Using a relevant theory is helpful for applying some parts of the theory into the proposed theory. This needs reviewing the past literature and the question here is that how the literature should be reviewed? [43]. The most important note about this action is to use what we call “relevant” literature. A definition for a literature review can be expressed in this sentence: “the selection of available documents (both published and unpublished) on the topic, which contain information, ideas, data and evidence written from a particular standpoint to fulfil certain aims or express certain views on the nature of the topic and how it is to be investigated, and the effective evaluation of these documents in relation to the research being proposed.” [61].

Regarding our research questions and also implementing the proposed method, we reviewed relevant books and articles which were carefully selected by recently cited sources and authors.
Taking into consideration the role of literature review which is to develop theoretical framework and also conceptual models, the act of combining relevant elements from earlier studies is helpful [43]. Regarding this matter, we have motivated our work by paying attention to the researches done in this field.

This section helps to position the problem which was defined in the beginning of the research and also helps understanding the concepts in similar projects as a guideline in implementation of our new system. We reviewed many books and articles which we have referred them in the references section. These sources of information were achieved from broadly cited authors within the field of semantic roles and biomedical.

In fact, according to Ghauri and Gronhaug [43] suggestion, the other roles of literature review can be listed as below:

- Structure the problem of research
- Recognise relevant concepts, methods and facts
- Using the existing knowledge to be involved in the new research

Identifying the advantages of literature review had encouraged us to use it as a research method. The way we have used this method and applied it in our study, is described below:

In order to getting benefit of relevant researches, we searched for different kinds of materials which fit into our research area and used the most cited ones out of them. Google scholar and the school’s library website helped us a lot for issuing this matter. Among these amounts of sources, we only used those with more relevant titles and those were more accessible. Studying researches about “FrameNet”, “PropBank”, “SRL”, “software development methods” and more on, gave us the understanding of concepts which we have reflected them in theoretical background section.

Finally by reviewing different structure of systems in the field of biomedical, we developed the framework of our system which is illustrated in Figure 4-1 and later described in chapter 4.
3.2 Suggestion

Next step in providing a research is “suggestion” phase. This phase is necessarily used after recognizing the problem of research’s field and can be applicable after making a proposal as a output of the problem recognition [59,60]. Suggestions are the approaches including methods and methodologies which help the proposal to solve the mentioned problem. Problems in a software system complexity can be solved by the approach of software development with focus on operation support systems, automation of the maintenance function, and development of a high-level programming environment [59, 60]. In design science research, a tentative design is essential as a part of the proposal. Tentative design is “Tentative design is an essentially creative step wherein new functionality is envisioned based on a novel configuration of either existing or new and existing elements.”[59, 60]

As we said above, suggestions are consist of approaches which may be methods and methodologies. Comparing with our research field, we have stated the problems as research questions and in order to deliver them as an output of our research, we obviously felt the lack of method.

By having suggestion’s definition in mind, the task of choosing an appropriate method reached us to the suggested method by He Tan. We expressed that method by implementing a system. The fact is that the method is developed by He Tan [1] which is based on theory of frame semantics. Using this method has led to a corpus labeled with semantic roles for the domain of biomedicine. We have supported the new method by means of developing a supporting environment. The suggestion was to deliver a new system to support semi-automatic labeling of the new corpus built by this method.

By defining the goal as “delivering a system with specified functions”, we decided to use Java as our programming language, and JavaNetBeans as our programming environment. In the suggestion phase we have described about development of system and the methodology we chose in next sections which are mainly mentioned in 3.3 and 4.2. According to this step, we recognized the need of requirements within system’s developing. For this matter we defined an activity titled by writing user stories, where the story clarifies next steps in the process. In fact these user requirements are objectives of our main solution which is delivering a supporting environment to support the method of frame semantics. Developing the suggestion’s step is the next phase which we have explained below.
3.3 Development: XP Methodology

This phase focuses on the development and implementation of the tentative design which was described previously in suggestion phase. Creative efforts are needed while moving from tentative design into complete design requires. Developing and implementing approaches are different due to the differences in making artifacts, sometimes an algorithm is needed in order to build the development technique [59, 60].

In our thesis work, we developed an environment for supporting the new method of corpus construction using frame semantics to express the meaning of natural language. Development environment is NetBeans IDE (Integrated Development Environment) and development programing language we chose is Java.

This master thesis project has been enthused by Extreme Programming (XP) methodology in system development process, which was chosen by us. In this section we have reviewed XP methodology concepts and also we have stated why it fits our field of work. Extreme Programming is based on values of simplicity, communication, feedback and courage. Following XPs values, shall lead to more responsiveness to customer needs than traditional methods and creation of software with better quality [62].

XP describes four basic activities that are performed within the software development process: Coding, Testing, Listening and Designing [62]. Each of these activities is described below.

Coding: The advocates of XP argue that the only truly important product of the software development is code that a computer can interpret. All code is produced by both of us, each programming one task on one workstation (pair programming). Each is responsible for all the code and is allowed to change any part of the code. We used the code standards in Java to make the code easier to read and understand by other students.

Testing: Acceptance Tests were held to verify that the requirements are easily understood and satisfy user’s actual requirements during regularly meeting. We always work on the latest version of the software, and upload our latest changes often. Test-driven development has Chosen to ensure that all code is properly tested before integration.

Listening: We as programmers must listen to what the customers expect the system to do, what “business logic” is needed. In order to communicate between user and us Planning Game has been followed. Planning is divided into two parts, release planning and iteration planning. In release planning the user and developers plan which requirements that shall be included in coming releases. Iteration planning plans the tasks for the developers and is done by the developers and has been done by us and our supervisor.

Designing: If all the previous activities are performed well, the result should always be a system that works. But in practice, this will not work and we
cannot avoid designing. Simple design was chosen to make the code easier to understand by others.

Comparing to our system, the requirements were defined in the beginning but later it was decided not to implement it based on ontologies. Figure 4-1 shows the general architecture of our system based on requirements. “Knowledge provided by ontologies” is considered in the architecture but in fact it is not implemented by us and we have not integrated the system with ontologies. As a result, the requirements will change again due to the use of domain ontologies. Considering what we have said about XP, this kind of methodology can better support the process of extending the system if the requirements change. By choosing XP methodology for this project, the possibility of using ontology can be applied to the current system because of the lower cost of change and possibility of improving code in JAVA language.

Using extreme programming (XP), we start to collect user stories (described in section 4-2-2) and conduct simple solutions in the first three weeks. Then we plan to have a meeting for release planning including our supervisor (fulfilling both user and client roles) and us (developers) to create a schedule that leads to every week meeting. After that we start our iterative development with that iteration planning which everyone agrees on it.

We looked for a methodology for software development after the system got in some trouble. Our requirement specification seems useless. Another problem that we faced was changing requirement that leads to recreate the schedule. All these caused to use XP. We solve the problems first by replacing user stories instead of requirement specification. Then we try an iterative way in development process. We use unit tests for the integration bugs as well as the acceptance test for production bugs. While both of us as a developer own the same core classes in programming, we could become bottlenecks for each other. So we try to make change to the core classes whenever there is a need by applying pair programming. Continuing this way, we add some other practices. We try to talking about problems and solutions to provide an open space to encourage each other and improve communication. And finally our problems have been solved and we got our project is completely under control. We show different features of the XP methodology in our thesis work [63]:

- **Spike Solutions: Simple and Focused Answers**

When we faced programming or design problems, we try to build spike solutions to explore answers. A spike solution is a simple solution program to figure out potential solutions. Most of the time they are good enough to only address the current problem without consideration other issues, therefore we expect to ignore them. The main idea of creating spikes was to decrease the risk of a programming problem together with increase the reliability of user stories [63]. It was helpful when a technical problem happens as a threat to hold the development process; both of us reduce a potential risk by working as a pair.
under examination and ignore all other concerns. Most spikes are not good enough to keep, so we expect to throw it away.

- **User stories: user requirement specification**

User stories are used to know what the customers expect from the system instead of the large requirement specification [64]. They serve as time estimation for the release planning meeting as well as the creation of the acceptance tests. Comparing to traditional requirements specifications, they only provide enough detail information to estimate how long the story will take to implement by developers. Another important issue is that there are no details of specific required technologies and algorithms, since the focus of stories is on user needs. We have met our supervisor as a customer regularly to receive description of the requirements face to face, when it was the time to implement the story. Each story gets at least 1 week to implement depending on the level of the tasks. User stories are described in details in section 4.2.1.

- **Release planning: on time planning**

The basic idea of release planning is that every project is quantified by four main things: scope, resources, time, and quality [64]. Scope is how much work is to be done. Resources are how many people are available. Time is when the project or release will be done. And quality is how good the software will be and how well tested it will be [64]. We can have a better control of all these issues by applying a release plan but no one can control all of them. The reason is simple: when you change one, you cause another to change.

It is important to make technical decisions while starting implementation, so a release planning meeting is used to lay out the overall project and build a release plan [64]. We try to set some rules to define a method for scheduling the priorities. Every user story need to be estimated by us on how long we need a programming. We come to the acceptance of for the first release 4 weeks of programming completely with nothing else to do. Then customer (our supervisor) has decided that which user story is more important and which is less. For example, adding an automatically POS tagger has a less priority in Release planning than manually tagging in the annotation task. Together we and supervisor decided on the set of the stories that need to be implemented as the first or the next release. We had two alternatives which plan must be based on: time or scope. We plan by time that 20 stories can be implemented before 1st October. We multiple the times in number of iteration to determine how many user stories can be completed in a given time of thesis work which is 5 months. These estimations are used to iteration planning meeting.
• **Iteration planning: add agility to development process**

An iteration planning meeting is called at the beginning of each iteration to produce that iteration’s plan of programming tasks [63]. Each iteration was 1 to 2 weeks long. User stories are chosen for this iteration by our supervisor from the release plan in order of the most valuable to the customer first. Failed acceptance tests to be fixed are also selected. The customer selects user stories with estimates that total up to the project velocity from the last iteration. The user stories and failed tests are then broken down into the programming tasks that will support them. Tasks are written down on index cards like user stories. While user stories are in the customer's language, tasks are in the developer's language. Duplicate tasks can be removed. These task cards will be the detailed plan for the iteration. We sign up to do the tasks and then estimate how long their own tasks will take to complete. It is important for us who accepts a task to also be the one who estimates how long it will take to finish.

• **Acceptance test: regular testing system**

Acceptance tests are created from user stories [63]. During iteration the user stories selected during the iteration planning meeting will be translated into acceptance tests. The customer specifies scenarios to test when a user story has been correctly implemented. A story can have one or many acceptance tests, whatever it takes to ensure the functionality works [65].

Acceptance tests are black box system tests. Each acceptance test represents some expected result from the system. Customers are responsible for verifying the correctness of the acceptance tests and reviewing test scores to decide which failed tests are of highest priority. Acceptance tests are also used as regression tests prior to a production release.

### 3.4 Evaluation: Data Collection and Survey

Evaluation is considered as an activity in software engineering to determine the quality of the proposed software [66]. After developing the proposal, the output should be evaluated. The evaluation phase focuses on evaluating artifact according to the set criteria that are always implicit and frequently made explicit in the proposal or awareness of the problem phase [59, 60]. Extra information achieved from development and results from running of the artifact are again collected together for another round of suggestion [59, 60]. The attention of evaluation phase is judging results according to performance and measurement of algorithm or design technique used in development phase.
Two phases which compose evaluation are: pre-study and evaluation of study. Pre-study phase is mainly about data collection which deals with collecting data from interest groups. Interest groups are those people who hand over the data for evaluation phase. Concerning different goals in evaluation, interest groups may differ [66]. Evaluation model can be shown in figure 3-1:

![Evaluation Model](image)

Figure 3-1: Evaluation Model

The model describes the step we have done during evaluation in order to get results. Pre-study phase defines the problems by help of interest group who played a role in collecting data. Reviewing the results by evaluation phase, make an action plan composed by different activities. These activities can improve the results effectively.

In this project, the interest group collaborated with us are experts in the field of computer science (especially in semantic roles area) and having knowledge in biomedicine aspect. Data is collected through surveys shown further in figures 5-11 and 5-12.
In majority of thesis projects in order to answer research questions, researchers need to decide what kind of data collection method should be used to gather some primary data. There are several options for collecting primary data: observation, experiment, interview or survey [43]. Decision on what kind of these methods would be used depends on the research problem and the research design. For some particular research problems, a researcher has to gather some specific information from an individual respondent to carry out analytic investigation. The communication approach should involve surviving domain experts and recording their responses for analysis. In these cases the survey approach can be a best choice of data collection methods which convinced us to get benefit of surveys for our system’s evaluation.

Survey is a method of collecting data that develops questionnaires or interviews for recording responses [43]. The excessive strength of the survey as a primary-data collecting approach is that abstract information can be folded by questioning people. Once the researcher has determined that survey is the appropriate method, the research problem will determine which type of the survey should be undertaken. The main types of surveys and questionnaires are descriptive or analytical [43]. With descriptive surveys we can identify a phenomena which we wish to describe while analytic survey are concerned with testing a theory. However both types of surveys are often used to identify the population. The population provides all the responses in order to answer research questions. The next important issue is to construct a questionnaire. Researchers should know what information they need and who should be the respondent.

In our thesis project, after the research problem has specified and an appropriate research design has developed, the next step was to select data collection instrument. Section 5.2.2 will explain more about survey method and also the results we got through this way of evaluation.

3.5 Conclusion Phase

The conclusion phase is the last step in creating a design science research. The results are focused to address the usage of new method of corpus construction. The main involvement of the conclusion is to achieve results, which are defined clearly in the purpose or objective of the proposal. We conclude after the evaluation phase from the domain experts and knowledge mentors, that the results are authentic and that they are truly mapped according to the purpose of this thesis.

The analysis of results, taken from surveys and available data leads us to have an overall understanding of system’s usage. It can be concluded that how accurate is the supporting system according to the percentage of system’s response to the user requirements.
4 Framework and System

We present a general framework for semantic role labeling. The framework combines a semi-automatic way of annotation with a frame specification option which motivated by an effective approach of corpus construction. Within this framework, we study the role of labeling sentences in biomedical applications.

4.1 Framework

The overview of the system is illustrated in figure 4-1, from the initial frame description to the construction of the annotated file. In the sections that follow, we will describe each step in detail.

Figure 4-1: General framework of system

Frame Specification

We first create a section where user can define and edit a frame, including a list of frame elements developed by domain ontology. Based on the list of the frame elements, a set of tags is prepared to use in the annotation process. A detailed description of this phase is shown in figure 4-2.

Selection of Examples

It is the first step of the annotation process. User can select an example sentence from existing options or input a new sentence.

Tokenization

Here the selected sentence is divided into tokens which are shown in the annotation table. One can change any token if is not agree on the way of tokenization.
Annotation
Annotators select frame element, phrase type and grammatical function for each token in the sentence. A more detailed discussion of this process can be found below in figure 4-3.

Save Formatting
All the result of the annotation and initial sentence are saved in a text file.

Figure 4-2: "Framing" Process overview
Since the system consists of two portions, the lexicon of frames files and the annotated example sentences (corpus), the process is similarly divided into framing and annotation.

### 4.1.1 Framing

The process of creating the frame files, that is, the collection of framesets for each target word, begins with the examination of a sample of the sentences from the corpus containing the word under consideration. These instances are grouped into a single frameset. To show all the possible syntactic realizations of the frameset, many sentences from the corpus are included in the frames file, which are called example sentences.

In some cases a particular sentence will not be attested within the related frame corpus; in these cases, a constructed sentence is used, usually identified by the user who wants to annotate the sentence. Care must be taken by user during the framing process to make allied sentences have the same framing, with the same number of roles and the same descriptors on those roles.
4.1.2 Annotation

We begin the annotation process by running a POS tagger on the corpus. This tagger incorporates an extensive lexicon, which currently encodes in java. Although tagger achieved high accuracy on data, the output of this tagger is then corrected manually for defining better mappings between grammatical and semantic.

Annotators are presented with an interface which gives them access to both the frameset descriptions and the full tokenization of any sentence and allows them to select tokens in the annotation table for labelling as arguments of the predicate selected. For any sentence they are able to examine both the descriptions of the associated frame and the example tagged sentences, much as they have been presented in the system. The tagging is done on a token-by-token basis, rather than all-words annotation of running text.

For new sentences, annotators had to determine which frameset was appropriate for a given usage in order to assign the correct argument structure. These sentences are arrayed in a classic file distribution, and their annotation were stored in a stand-off notation, referring to frames within the system without actually replicating any of the lexical material or structure of that corpus. Both role labelling decisions and the choice of frameset were adjudicated by an annotator.

The annotators themselves were drawn from a variety of backgrounds, from undergraduates to holders of doctorates, including linguists, computer scientists, and others. Undergraduates have the advantage of being inexpensive but tend to work for only a few months each, so they require frequent training. Linguists make the best overall judgments although several of our non-linguistic annotators also had excellent skills. The role labelling process curve for the annotation task tended to be very dependent with annotator’s background but to become annotators becoming comfortable with the process does not take more than one hour of work.

4.2 Method of System Implementation

4.2.1 User Requirements: User Stories

A user story is a form of software user requirement that has become quite popular in Agile Methodologies such as Extreme Programming and Scrum. Unlike more traditional methods such as a System Requirements Specification or Use Case Diagrams, the emphasis in these methodologies is simplicity and changeability. Therefore we have designed user stories to be easily described and understood, and more importantly easily changed by the end user during the project [67].
User stories are short, simple description of a feature told from the perspective of the person who desires the new capability, usually a user or customer of the system. They typically follow a simple template [67]:

As a <type of user>, I want <some goal> so that <some reason>.

In our system four main user stories are listed as below:

- As a user, I want to add a new frame including name and definition.
- As a user, I want to modify predefined frame definition but not the associated frame elements.
- As a user, I want to add new example sentence and annotate it.
- As a user, I want to edit an existing example sentence and annotate it.

It is quite difficult to get a large number of User Stories at once. What happens is that you get them from time to time. We have decided to get the User Stories, Analyse them, Design and Document, then implement and test. Proceeding in this fashion we had iteration reports, which helps to complete and updated information about the system and its features as the project progresses.

4.2.1.1 User Story 1: Scenario

Objective is adding a new frame to the corpus. Scenario will start through “Frame Definition Section” and moving to “Frame Name and Frame Elements” tab in the application. User can define the frame in terms of name and save the changes in a file. The story will be continued by choosing “Frame Definition” tab where user can set a definition to the desired frame located in dropdown list.

4.2.1.2 User Story 2: Scenario

Objective is modifying predefined frame definition without changing any other related frame elements. Scenario will start through “Frame Editing Section” and loading the desired xml file or dataset as corpus. The story is continued by choosing the desired frame among other frames located in the left side table. Clicking “Edit” button will load all the associations and relations in the right side table. User can change the content of fields he wants. In this case, the target field is “Frame Definition”. The modification will be applied by clicking “Save” button.

4.2.1.3 User Story 3: Scenario

Objective is adding a new example sentence to a frame. It is assumed that user wants to use a pre-defined corpus and add an example sentence to a pre-defined frame. Scenario will start through “Frame Definition Section” and loading the desired xml file or dataset as corpus. In this panel user can select the frame which the new example sentence will be added to. Writing the example sentence and clicking “Save” button will answer to the user’s need.
For annotating the example sentence scenario will continue through “Annotation Section” and loading the desired xml file or dataset as corpus. In this panel user can select the desired example sentence for annotation task by choosing the proper frame. Clicking “annotate” button will tokenize the sentence and load them in table. Selecting each row of table gives the opportunity to edit or delete that token. Editing is done by clicking “Edit” button, choosing proper roles and at last clicking “Enter” button. This is applicable to all tokens and by clicking “Save” button the result of annotation task will be saved in a file by user. User’s annotation result can be compared to POS Tagger’s annotation result by clicking “Tagger” button.

4.2.1.4 User Story 4: Scenario

Objective is editing an existing example sentence of a frame. Scenario will start through “Frame Editing Section” and loading the desired corpus. All the available frames in the corpus will load in the frames table. User can select the frame and load its related example sentences by clicking “Search” button. This user story will be finished by modifying example sentence and clicking “update” button.

For annotating the example sentence scenario will be continued as we described in user story 3.

4.2.2 Software Development Environment

4.2.2.1 Programming Language for Development

First of all, the object oriented programming language should be chosen for realization of described system. Selection of programming tools is one of the most serious steps in development cycle and will depend on few points mentioned below [68]:

- Speed of programming development
- Convenience of user interface
- Possibility to change/improve the code easily
- Performance of software
- Possibility to create useful documentation automatically

Based on the advantages listed below, Java programming language has been chosen for the practical implementation [69]:

- Portability: Java runs on most of the hardware and software. Theoretically, by using this language, we make the system compatible with other platforms.

- Reliability: Java runtime realizes multiple checks of byte-code to avoid any inconsistency and to verify correctness of code. As compared to C
and C++, some features were deleted from Java language (like pointers and automatic type conversion).

- Support of multithreading: This feature can be defined as “the ability of a program or an operating system process to manage its use by more than one user at a time. It even manages multiple requests by the same user without the need to have multiple copies of a program running in a computer. In our case multithreading can be used as an advantage for GUI creation.

- Robustness: It includes early checking of possible errors. Java compiler is able to detect many problems. Java realizes runtime exception handling feature, which means that it can catch and respond to an exceptional situation so that the program can continue its normal execution and terminate gracefully when a runtime error occurs.

As a final point, by technical estimation made before the programming realization, advantages which Java tools provides to us are more ponderable than other tools.

### 4.2.2.2 Integrated Development Environment

Among different integrated development environment, we have found that NetBeans platform fits our needs. It is a cross-platform open source IDE for Java that comes with a syntax highlighting code editor that supports code completion, annotations, macros, auto-indentation, etc. It includes visual design tools ( wizards) for code generation. It integrates with numerous compilers, debuggers, Java Virtual Machines and other tools.

Before the final acceptance of chosen IDE, small research and comparison was done on their possibilities and properties. Since NetBeans supports more modular structure than Eclipse IDE and IntelliJ IDEA, we have decided to pick it for the development process.

### 4.2.3 Interface Design

Developed software contains graphical user interface module, which allows end users to define new frame or to annotate. To create GUI in Java we will make use of the swing package called javax.swing. This package contains a lot of classes that will create GUI components for us.

Our first objective will be creating a frame. Once we have the frame, we can add other components to it. Next is to look at what we want to achieve. There is a title, it has a specific size, you can see it visible and you can close it. Knowing this, we need to check what methods are in the JFrame class.

The container is the area that we can put the components like buttons. To get access to this container we make use of the method getContentPane() from the
Framework and System

Jframe class. To configure the layout of the container area we use setLayout() method. Components are added from left to right and top to bottom when we make use of the gridlayout.

We have added some labels by using pane.add(label). We make use of the JTextField to be able to enter some data by user. To give the user the capability to start an action, we will give the user a button. For this component we will use the JButton class and to create the component as a button we will send a text as a parameter that will appear on the button.

4.2.4 Java Class Design

In this section short description of developed package and classes to realize the functionality is described. As mentioned before, two major classes named Framing and Annotation are defined in the package.

Annotation class includes the table for annotating an example sentence. This will be done manually by user. We write an algorithm to tokenize the sentence which user wants to annotate. It allows user to change the divided tokens if he is not agreed with the result of the tokens which are written to the tables in rows. The output of the process is saved in the text file with the related frame.

Creating a new frame and all the related frame elements is made possible in Framing class. You can add, remove, or edit frame definition and frame elements through it. There is also a need for defining example sentences and relates them to the frame definition which is done in this class. The output of the class with all the relations will be saved in a XML file to easy access and read. Framing class mainly uses TransformerFactory class for the “save” method and DocumentBuilderFactory class for creating XML file. The data of this XML file will be filled by whatever user defines in Frame Definition Section of the system. This file will be also updated when user updates some fields in Frame Editing Section or Frame Deleting Section. This process should obey a code in order to add desired tag names and data to the XML file. For tagging or in other words labelling the nodes in XML file, solution is to consider the input of each textbox as a TextNode. The related elements are then created for appending to the node. For adding data to the nodes and basically to each tag, we have used the method getElementsByTagName.

Finally the design of the classes represents a compromise between the need for user to annotate the sentences, find the related frame, and define frame and elements at one hand and elegant object oriented design at the other hand.

The system can be started by clicking on the jar file and then the GUI of the system will be appeared. Detailed about GUI and functionality of the system has been described in section 5.2.1.
4.2.5 The Corpus Database

The considered corpus should support a structured collection of data as a database in order to be retrieved when needed. This database can be loaded through the system which we have developed or a new corpus can be saved with the same database structure. In this work, the data including information on frames, frame’s related associations and also annotated sentences are organised in two ways of files as database which we have described below:

4.2.5.1 XML File

As we said, the data including information on frames and its related associations are stored in an XML file. XML refers to Extensible Markup Language which explains a class of data object called XML documents. XML documents follow two structures which the first is physical structure and discuss the document is made by entities. The second structure is logical one discuss that document is made by declarations, comments, elements, character references and processing instructions. The importance is composing a well-formed XML documents by these two structures [70]. XML data can be better explained by different XML schema languages such as XML DTD, XML Schema, XDR, SOX, DSD and etc. [71]. A comparison between these schemas is shown in figure 4-4 below:

![Figure 4-4: Comparison between Different XML Schemas](image-url)
We chose using XML DTD for these three reasons:

1. It has been survived for a long time and as they are supported by considerable organisations, they have a high chance to be continued using in future as well.
2. Known applications using this schema language
3. It is easy to learn despite its use of proprietary syntax

Figure 4-5 shows a piece of XML file used as corpus database.

Figure 4-5: XML file containing data through frame semantics

4.2.5.2 Text File

A text file is considered as database for annotated sentences. User chooses to save the database himself and the file will store the data in a structure starting by original example and following by annotated tokens. Figure 4-6 shows a piece of this file.
4.2.6 System Requirements

The prerequisites for using this system are not very limited. Anyone who finds the system effective to his work and related to his domain of interest can work with the system. In other words, anyone whose work is related to biomedical text mining can use this supporting system in order to recognise biomedical terms with annotations. There is no limitation in installation of the application as we have made a jar file which can be run on every computer updated by a java runtime environment. The only considerable issue is that corpus’s structure is in the format of XML which means that XML document should exist for loading the corpus into the system. The work is delivered in a package containing a jar file and an XML file as corpus database for testing the system. We have tested the system in an environment equipped moderately with at least these features: any kind of OS, processor 500MHz or higher, available disk of 100 MB, Memory 256 MB. It is expected that every computer having this minimum features, supports the system properly.
5 Results and Discussion

This section presents the results of our work which we have done during this master thesis. It also states how the objective of this thesis work is reached according to the development process. Once the objective is achieved, the result is ready. Objectives are those explained as research questions as well as figuring out as user stories explained in section 4-2-2. In other words, the result is a system for building corpus annotated with semantic roles. As we described about the purpose of this thesis report in section 1.2, the development of a supporting system is done based on a corpus of biological transport events which the method of building corpus was previously proposed by He Tan.

For better understanding and also delivering more clear explanation of the results, we have categorised results in 2 types as theoretical results and practical results and described them in that way. In this format, results can better describe the specific research question. We have reviewed various similar works as a constructive research by means of literature review which leads us to explain the theoretical findings. Achieving practical results in a systematic way has got benefit of DSR method and for the development of them, we have applied some development tools like JavaNetBeans IDE version 7.2, using Java codes and Java libraries for the execution of proposed solution. These results are more described in details in the following parts. System requirements are defined in section 4-2-1 and also we explain how the system supports semi-automatic labelling of the corpus construction by means of screenshots in section 4-2-5. Afterwards the analyses of the results are discussed by means of two surveys which are fitted to evaluation section.

5.1 Theoretical Results

The first research question “Investigating how to support method of building corpus annotated with semantic roles?” is finely answered in this section as well as second question “How the general framework is developed”. These types of results are the output of our research through literature review and also reflected in theoretical background section. According to the understandings from different projects, there is a need to solve the problem for lack of large corpora and also solve ambiguities in possible meaning of terms for a domain such as biomedical domain. We came to the decision to get benefit of frame semantics to easily consider all definitions and contexts.
Describing all possible events can be done by means of ontology because ontology can define entities and relationships in a domain completely. As a result, the system should have the ability to support parsing and visualizing lexical properties of ontological terms, defining frame semantics description and annotation task based on such method. Finding out these features forms the theoretical result of the work. For this matter, a general framework is proposed which is described in details in section 4-1. General framework considers knowledge provided by ontology for injecting information using frame semantics. This framework includes two phases of framing and annotation illustrated in two different flow charts as figure 4-1 and figure 4-2.

5.2 Practical Results

In this section, the third research question “How to implement a system based on the framework” is finely answered. Our practical results can help us to compare the efficiency of the system through supporting method of frame semantics.

5.2.1 Implementation Results

In this section we explain the functionality of the system by means of figures which are in fact Graphic User Interface (GUI) of the system. As a result, it is understood how this system supports the user’s work in an easier way. Discussing about the needed components, lead us to remember again the structure of method and supporting system. So again we can refer that the definition and description of the frames in the corpus, which is a part of this system’s task, strictly follows the domain knowledge delivered by the Gene Ontology describing the events. The core structure of the frame is similar to FrameNet’s structure. The scenario’s explanation evoked by the frame is delivered, besides a list of the FEs and their definitions.

By means of this supporting environment, user has the ability to work through frame semantics and manage tasks such as: add a new frame with a name and description, add frame elements with their name and descriptions, and add example sentences to the new frame. It is also possible to edit a frame and its associations in a corpus through system. The stored data is presented in semi-structure schema by human. In other words, our system saves the definition and description of the frames and the related items in an XML file. This was discussed before in section 4.2.5.1 with presenting figure 4.5 where all possible items are set into “FrameDefinitionSections” tag consisting of different subtags and sub-subtags. Figure 5-1 shows the user interface for defining new frame and its definition. Other figures regarding different functionalities are presented in appendix 1.
System can make output through two ways. One way is when the user selects the sentence and annotates the divided tokens separately. In this way, annotation is done manually by user. System works with the defined frames in it, and annotates the sentences given by user. In fact the output of the system is a text file containing annotated tokens which are derived by saving the annotations labels to each token. This was previously shown in figure 4-6. The other way is annotating by help of tagger. In this way, a POS Tagger is responsible in suggesting the labels to each token. User then can use this suggestion and apply it to his annotation work and then save the result. The text file will be filled by the sentence itself as well as tokenized parts again. For this way, we added a java library in our coding which delivers the POS Tagger. The tagger used in this system is accessible by adding “Stanford POS Tagger”.

Figure 5-1: Adding new frame and its definition
5.2.2 Evaluation Result and Discussion

Once the implementation of the system has finished, we have identified data collection method for the evaluation of system. We could record the verbal behaviour of system users by an effective tool to get opinions called survey which is the most popular method for measuring expectations [72]. When our research problem has formulated and the study has clearly defined, this governs the type of survey we should have used. The focus of the survey is more on a representative sample of the system rather than an analytical survey. Therefore using a descriptive survey was more appropriate since we wanted to obtain user attitudes towards working with our system. To construct the survey, a review of earlier literature was a key role to determine what type of questions has to be included in the questionnaire. In the development of survey, first we specify what type of information is required to get the feedback from users. Second, we consider only domain experts should be our respondents. To do that, we have contacted two experts in the field of biomedical domain. Finally we consider some guidelines for the construction of the questions. For example questions should be simple, concise, realistic demand, specific and etc.

Applying this method needs some questionnaire. Various surveys are created to deliver the results because different types of goals need different type of questions. Considering this matter, we created two surveys; one is regarding the goal of easiness, and the other one is considering the goal of speed. Our surveys are created by the help of GoogleDocs, and then the survey’s links are published through internet among different users. The task is working with the system and filling the questionnaire as a feedback. Designed surveys are shown as figures in appendix2. Figures 8-10 shows the survey based on the goal “easiness” and figure 8-11 presents the survey for analysing the system for the goal “speed”.

Filled surveys are analysed by the survey provider which in this case is GoogleDocs. Analysis result can be shown in the format of charts and graphs showing variety of feedbacks by percentage. Derived results from surveys show to what extend system is easy in meeting the needs of users. Below the general results are shown. Figure 5-13 is a pie chart which shows the percentage of satisfactory in aspect of easiness and figure 5-14 is a bar chart which shows the percentage of satisfactory in speed’s aspect.
Results and Discussion

Figure 5-2: Evaluation Result Regarding Easiness of System

Figure 5-3: Evaluation Result Regarding Speed of System
Comments achieved from interest group, caused the creation of action plan which discuss about possibility of improvement. This action plan consists of action proposal such as:

- Instead of manually starting the tagger for producing suggestion, automatically fill in the POS suggestions produced by the tagger
- Removing the need of running system repeatedly in case of re-opening the corpus or opening another corpus
6 Conclusion and Future Work

This thesis work can be seen as a step towards a better integration of theoretical method of corpus construction and practical semantic role labelling. In this paper, we support a method which aims to build corpus that are labelled with semantic roles for biomedical domain by proposing a general framework. The proposed framework can be tailored to any domain, and it is not prescriptive of particular tools and techniques. Based on this framework, a system is developed to support this method.

The system comprises a biomedical corpus, organized in terms of semantic frames in ontology-based domain knowledge. In particular, we develop a supporting environment which has the components that allow the user to define frame semantics description, frame elements, add lexical units and example sentences. The system also supports retrieving defined example sentences in order to annotation task, annotating sentences and saving them in specific format. It is available as a software environment for building corpus annotated. While it includes support for common frames in biomedical domain, it is also easily extended. Existing frames can be augmented and new ones added within the features that system provides.

Developed system has an annotation scheme for biomedical texts and produces an associated corpus. The corpus is unique within the biomedical field in that ontologies act as a basis for knowledge creating domain-specific semantic frames. It is hoped that the corpus will boost research into other areas of domain-specific SRLs. However, our results show that use of a human interaction in semi-automatically systems may cause some difficulties. This is possibly due to the different expertise levels of annotators within the expression domain. A solution for this would be to analyse the domain knowledge of annotators in greater detail and, where appropriate, provide extra training in the assignment of tags in the annotation task. Besides, validation of the system is predicted by applying the system to texts from different biomedical domains and by using data collection methods for evaluation of the results.

As we described in general framework of our work, the system can be integrated later with ontological knowledge. Following our results, future work could possibly considering an ontology plugin for the integration of system with ontologies. Another subject for future work can be explained as choosing alternatives between taggers. In our project, we have only got benefit of Stanford POS tagger. This part of the work can be later replaced by a plugin for generating different annotation suggestions by using various POS tagger.
7 References


References


Appendix

Appendix 1: Practical Results (Screenshots of system)

Figure 8-1: Adding frame elements and definition to a frame

Figure 8-2: Editing a frame and its related options
**Figure 8-3:** Selection sentence for annotation purpose

**Figure 8-4:** Confirming the selected sentence to be annotated
Figure 8-5: Tokenisation of sentence

Figure 8-6: Setting roles to tokens
Figure 8-7: Saving the result through filled table

Figure 8-8: Annotate using POS Tagger
Figure 8-9: Dividing tokens
Appendix 2: Surveys

Easiness

Please fill out this survey which gives us feedback on how easy the system is. Thanks!

1. In what level of satisfactory did you find “how to use” section?
   - Not at all satisfied
   - Slightly satisfied
   - Moderately satisfied
   - Very satisfied
   - Extremely satisfied

2. It was easy to install and store system in order to use it
   - Not at all satisfied
   - Slightly satisfied
   - Moderately satisfied
   - Very satisfied
   - Extremely satisfied

3. The interface was designed easy for performing system functions
   - Not at all satisfied
   - Slightly satisfied
   - Moderately satisfied
   - Very satisfied
   - Extremely satisfied

4. The system has all functions and capabilities I expect to have
   - Not at all satisfied
   - Slightly satisfied
   - Moderately satisfied
   - Very satisfied
   - Extremely satisfied

In case of missing a function, please write it below:

5. The system responded in a consistent and predictable way
   - Not at all satisfied
   - Slightly satisfied
   - Moderately satisfied
   - Very satisfied
   - Extremely satisfied
Appendix

In case of happening an unexpected response from the system, please write it below:

6. System provides message when a fault happens by user
   ☐ Not at all satisfied
   ☐ Slightly satisfied
   ☐ Moderately satisfied
   ☐ Very satisfied
   ☐ Extremely satisfied

7. The system functioned efficiently in working with input/output files such as: Loading a new input file, editing the loaded file as corpus, saving the edited file and saving the output file as annotated tokens
   ☐ Not at all satisfied
   ☐ Slightly satisfied
   ☐ Moderately satisfied
   ☐ Very satisfied
   ☐ Extremely satisfied

8. I could easily complete different tasks by using the system
   ☐ Not at all satisfied
   ☐ Slightly satisfied
   ☐ Moderately satisfied
   ☐ Very satisfied
   ☐ Extremely satisfied

In case of having problems in (a) task/tasks, please specify the task with a short description of problem below:

Submit

Powered by Google Docs

Figure 8-10: Survey "Easiness"
Appendix

Figure 8-11: Survey "Speed"