BENCHMARKING OF
SEMANTIC ANNOTATION SYSTEMS

by

Minal Patel

B.E. in Information Technology,

Gujarat University, India, 2011

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2014

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

BENCHMARKING OF

SEMANTIC ANNOTATION SYSTEMS

Minal Patel
Master of Science, Computer Science, 2014
Ryerson University

Abstract

In this research, an effort has been made to evaluate the semantic annotators with a systematic subjective evaluation technique. So far, most of the previous evaluation efforts have involved creation of gold standards and by measuring basic metrics, the performance of semantic annotators has been analysed. But in this work, a subjective evaluation technique has been applied to evaluate some of the publicly available semantic annotation systems. In this method, 60 participants have been involved in the evaluation. A survey has been carried out to collect the response from participants about what they think how well the annotators perform on different types of texts (e.g. long texts, short texts and tweets). Their responses have been analysed using standard statistical tests. Using this approach, it has been concluded that Wikipedia Miner performs better on long texts and Tag Me performs better on short texts and tweets than other systems.

ACKNOWLEDGEMENTS

I express sincere appreciation to my project supervisor, Dr. Cherie Ding, for her valued and generous guidance and help in the completion of the thesis. I am also very thankful to my Co-supervisor, Dr. Ebrahim Bagheri, for providing the expert knowledge. I would like to thank them for giving me the opportunity to work under them.

I would like to express my profound gratitude to my husband Nikul who has been a constant source of encouragement for me throughout my studies. To him, I dedicate this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviations	**Meaning**
BNC	British National Corpus
CoNLL	Conference of Natural Language Learning
CRF	Conditional Random Fields
CZ dataset	Cucerzan's dataset
D2W	Disambiguation to Wikipedia
KBP	Knowledge Based Population
KIM	Knowledge and Information Management
KW Test	Kruskal-Wallis Test
M&W	Milne and Witten
NER	Named Entity Reference
NIST	National Institute of Standards and Technologies
SAP	Semantic Annotation Platform
SPSS	Statistical Package for the Social Sciences
TAC	Text Analysis Conference
U Test	Mann-Whitney U Test

CHAPTER 1

INTRODUCTION

1.1 Background

Due to the rapid growth of technology in recent years, a huge amount of data, and hence information/knowledge, is at hand in the Web. These data on the Web, unfortunately, are only understandable by humans. Now it is a necessity to manipulate available web documents using machines i.e. computers and create machine readable documents. Tremendous amount of efforts have been devoted to developing systems that can create or manipulate any document using available web corpora and as a result, some semi-automated systems now exist for this purpose. These systems are able to provide additional information about any given textual document by identifying proper related keywords and linking them to the ontological concepts. These systems are called Semantic Annotation Systems or Semantic Annotators.

Figure 1.1 shows the basic structure of a semantic annotator. A semantic annotation system processes the raw textual document and by using a background ontology (framework for organizing the information), the annotation system annotates the document. The process of annotation involves two basic steps: (1) finding the keywords (called mentions) and (2) linking them to the relevant ontological concepts.

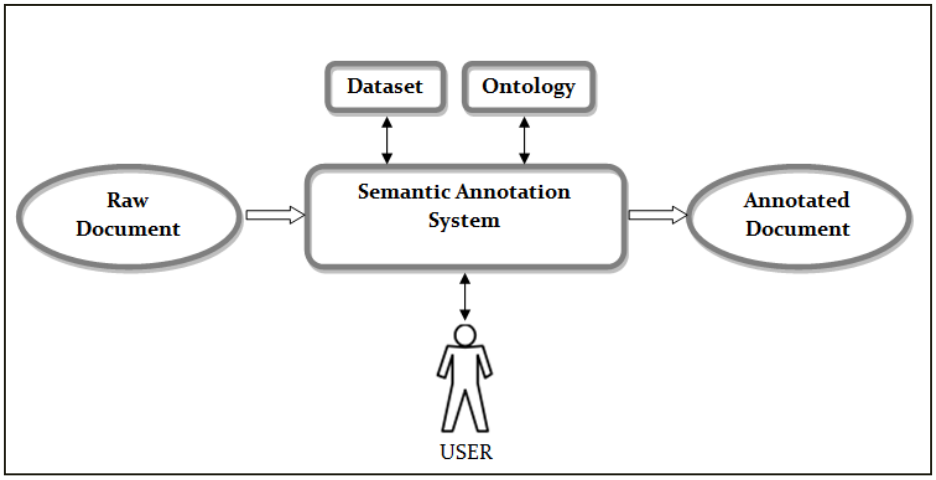

Figure 1.1 Basic Semantic Annotation Principle

The ambiguous terms – words that could have more than one meaning in different contexts – can also be annotated. One good example is the keyword "London". London is the capital city of England and London is also a city in Ontario, Canada. To pick the right one, usually the annotation systems check the surrounding context and links the ambiguous terms to the appropriate concept. For any system, the underlying meaning of the ambiguous terms depends on different aspects, e.g. the ontology and the, disambiguation algorithm used among other things.

In general, there are two key points for any semantic annotation system: (1) Speed and (2) Accuracy. It is important to identify mentions quickly, and disambiguate them accurately. A simple example, to understand this, is Wikipedia – the largest encyclopaedia available to date. Wikipedia provides information for any topic. These articles provide some keywords that lead the reader to their respective article pages. But in Wikipedia, the keywords are annotated and linked by humans. No doubt, Wikipedia provides accurate annotations, but it has involved many hours of human experts to manually prepare its documents. Semantic annotation systems do these automatically and take a very small amount of time, however, some struggle with accuracy. Some of them handle long texts accurately, some of them are designed for short texts and there are some annotators available that give great annotations for popular places, names, organizations, among others. In this thesis, an effort has been made to systematically evaluate the performance of some of the publicly available annotation systems.

The basic requirements of semantic annotations and its overview are presented by Uren *et. al.* [46]. Semantic annotation systems provide the relationship between concepts and a document. According to Uren *et. al.*, the semantic annotation task has the following four perspectives that are very important: (1) Ontology – a structural framework for managing available information, (2) Documents, (3) Annotation – Annotation links ontology to the document, and (4) User. These four perspectives might have one or more requirements, e.g. as per the ontology perspective, different frameworks should be supported and as per the user perspective, the documents should be easily created, read and shared among different users. Based on this, seven basic requirements are identified to design a powerful semantic annotation system. They are as follows:

- **Standard formats** – Standardized formats of ontologies and annotations are required wherever possible because a great amount of effort has been invested in developing and updating the resources. The standardization provides a bridge to the diverse resources which can be accessed simultaneously and hence, users and annotators can share information.
- **User-centred/collaborative design** – to provide collaboration between different field of experts or users, so that the documents can be created, read and shared easily. For instance, in the manufacturing world, the documents should provide enough information to both engineering technical experts and sales team (who may not know all technical terms).
- **Ontology support** – to provide support to different frameworks and to support appropriate ontology formats so that annotators can work with different ontologies.
- **Support of mixed document format** – On the web, the documents are available in many formats e.g. HTML, XML, CSV etc. The system should provide access to handle different document formats.
- **Document progression** – to give users an access to edit or update the annotations as the available documents change their versions.
- **Annotation storage** - to store the annotations and keep the consistency for newly created document versions.
- **Automation –** This should be used to finish the annotation task effectively and accurately.

The above mentioned requirements are the basic requirements in creating good annotation system. But the focus of this research work is kept only on the general performance of the annotation system. In short, in this thesis, only Automation of different semantic annotation systems has been compared.

1.2 Problem Statement

Different semi-automatic annotation systems have been developed in recent years. A few examples are: DBPedia Spotlight [9], TagMe [45], Wikipedia Miner [51], Wikimeta [50], Illinois Wikifier [22] and Denote [10]. The performance of these systems, in fact, most of the semantic annotation systems have been measured using one common approach. The evaluators have developed gold standards, i.e. human annotated documents. Whenever the authors made their

corpora or used the publicly available corpora, they also put efforts to manually annotate those corpora to create accurate standards to which they could compare the outputs of annotation systems. Usually, in the creation of the gold standard, more than one human annotator are involved and the mentions are selected whenever predetermined level of agreement takes place between the human annotators. The outputs of the systems are compared to the gold standards and the performances of the systems are measured by different metrics, e.g. Precision, Recall, F-measure, Ranker and Linker Accuracy (discussed in Chapter 2). These metrics basically count how many annotations the systems gave and how many were accurate when compared to the gold standards. This is currently a typical approach and new systems are often tested using the same approach. On the other hand, in this thesis, the semantic annotation systems have been evaluated differently. A systematic subjective evaluation technique has been applied. By doing this, human participants have been involved in the process of evaluation which could provide more accurate results. Involvement of many human evaluators can engage more complete and diversified viewpoints. This technique can provide what users think about the systems' outputs and this information can be used to improve the systems in some specific direction e.g. some systems give too many or too few annotations, some may work well only for specific type of texts and can be improved for other types of texts, etc.

1.3 Objective and Contribution

The goal of this thesis is to involve human participants in the evaluation of publicly available annotation systems. This has been achieved by performing a systematic subjective evaluation technique – a different approach to evaluate the semantic annotation systems. Also the benefit from this technique is that, "End Users" are directly involved in evaluating the annotation systems. So the direct responses from the users are accounted which could be beneficial for the future development of semantic annotation systems.

In this study, six different systems (DBPedia Spotlight, TagMe, Wikipedia Miner, Wikimeta, Illinois Wikifier and Denote) were evaluated by 60 users who were asked for their opinions regarding the performance of the annotation systems. The responses were gathered in the form of a survey questionnaire. The questionnaire included 13 specific questions targeting on the ability

of system to perform specific tasks. The responses about each question provide the details like how the users feel about the system and any possibility to improve the system for that particular task. These responses have been collected and analysed in detail to evaluate different systems for long texts, short texts and tweets. In our work, two types of variables were present: (1) Type of texts and (2) Different Systems. By performing Kruskal-Wallis Test, Mann-Whitney Test and Chi-Square Test, the result concluded is: for long text, Wikipedia Miner performs better than other systems while Tag Me shows better performance for short texts and tweets than other systems.

1.4 Outline

The remainder of this thesis is organized as follows:

Chapter 2 – Literature Review: This chapter outlines some of the previous efforts that have been devoted to developing and evaluating semantic annotation systems. It also describes the corpus and measures other authors have used to evaluate their work.

Chapter 3 – Methodology: In this chapter, the detailed procedure followed to perform the subjective evaluation has been explained. It provides the details of the approach, the questionnaire and the survey that has been used to evaluate the systems. This chapter also provides brief introduction of the six systems tested here. These six different systems have been tested using three different types of texts, i.e. long text, short text and tweets.

Chapter 4 – Analysis: Detailed step by step analysis is presented in this chapter. Two separate test cases are considered: (1) Type of texts as independent variables and responses from the participants as dependent variables and (2) Different semantic annotation systems as independent variables and responses from the participants as dependent variables. Since we have 13 questions, the analysis has been performed for each question for both cases. Different test methods have been applied to perform this analysis.

Chapter 5 – Conclusion and Future Work: This Chapter concludes by summarizing the work done in this thesis and provides recommendations for potential future work based on the current contributions.

CHAPTER 2

LITERATURE REVIEW

In this chapter, a brief introduction to semantic annotation systems provided. Moreover, this chapter review the previous related work related to semantic annotation systems and their evaluation. Any annotation system basically works on some common principle, but its efficiency depends on several factors, e.g. framework, database, baselines and algorithms used for disambiguation. This chapter covers some of the basic steps involved in the task of annotation. But since the goal of this study is to provide a new evaluation technique, this chapter concentrates more on the evaluation methods, corpus types and metrics used rather than algorithms and baselines.

2.1 Background of Semantic Annotation Systems

The Web has tremendous amount of information in various places and in various formats. These unstructured or semi-structured texts are only manipulated by humans. Semantic annotation systems provide the additional semantic contents to the web pages by annotating the keywords and linking them to the web articles. In other words, the semantic annotations make the web pages machine readable.

There are two types of semantic annotation systems available: 1) Manual and 2) Semi-automatic. So far, no fully accurate automatic system has been developed. Manual annotation is very expensive and also does not consider multiple sides of data sources. Moreover, the volume of the available web documents, required to be annotated or updated accurately, is very large. This could require more human effort. But semi-automatic annotation systems annotate the documents very quickly with least amount of human effort.

Semantic annotation systems work using the following two basic steps: (1) keywords extraction and (2) linking those keywords to the ontological concepts. This two-step process is called text annotation. The annotation ability of the system depends on the Semantic Annotation Platforms

(SAP) that can be classified into two categories: Pattern based and machine learning based. Pattern based SAPs can be sub-divided into two approaches: discovery and rule-based. [36] In the discovery approach, initial patterns are manually defined. Entities with the same patterns are searched continuously and new patterns are discovered. This process stops when it cannot find any additional entities or the user stops the process. The pattern-based approach can also be started using user defined rules. These approaches are useful only where there is no frequent change in the documents. On the other hand, machine learning based SAP can be sub-divided into probabilistic and induction approaches. Probabilistic approach uses statistical models to search the content while induction approach uses the learning patterns via language processing for information extraction. AeroDAML [23] [18], KIM (Knowledge and Information Management) [33] [34], MUSE [25] and SemTag [12] [11] are examples of pattern based SAPs and use rule-based techniques. Armadillo [13] and Ont-O-Mat PANKOW [5] use pattern discovery technique, while MnM [48] [47] and Ont-O-Mat Armilcare [20] are examples of the wrapper induction type machine learning based SAPs.

In many sectors, these semantic annotation systems have been successfully implemented. The most beneficial application, as stated earlier, is the semantic web creation. Because of the semantic annotation systems, the semantic annotations are now readily available in the web pages that provide machine readable documents [44]. Media management (BBC News), business and organisational management, Government data [26] and regulatory text management [37], data integration (Oracle Semantic Technologies), e-Commerce also use the semantic annotators and semantic web.

2.2 Previous Work and Evaluation Techniques

In the past, automatic augmentation of text with links to external web pages was introduced, but this effort was criticized by expressing some concerns that the pages were secretly modified for commercial purpose. Microsoft has shown some interest to introduce Smart-Tag service to automatically link text to other web pages within the Internet Explorer, but this effort was aborted too because of the same concern [32]. But Wikipedia – the largest publicly available

encyclopaedia – opened the door for many services since Wikipedia's sources are impartial and publicly available without any commercial purpose.

In 2007, Milhalcea and Csomai [30] developed the Wikify! system to annotate documents linking them to Wikipedia pages. This system works in two steps: (1) mention detection and (2) disambiguation. The first step, mention detection, identifies keywords or phrases from which the links should be made. This is done based on link probability approach. Link probability is defined as the number of Wikipedia articles having that phrase considered as an anchor (mention) divided by the number of Wikipedia articles having that phrase. Whenever this value is greater than some predetermined threshold value, that phrase is considered as a mention. The second step, disambiguation, links those considered mentions to the correct Wikipedia articles. For the ambiguous terms, Wikify! checks the surrounding text and detects the central idea and links to the correct article. In this work, the mention detection and disambiguation phases were evaluated separately. The British National Corpus (BNC) and the entire corpus of Wikipedia articles were selected as corpus. The performance of the system for BNC corpus was not great and hence, they reported the performance for the Wikipedia articles only. For the gold standard, they used manually annotated Wikipedia articles as well. Around 85 documents containing a total of 7,286 links were chosen as a test set. The keyword extraction methods were evaluated by comparing the keywords automatically selected with those manually annotated in the gold standard dataset. For the evaluation of the second phase, disambiguation, the same set of pages was used. Precision, recall and F-measure were used. Finally, the whole system was tested against 10 randomly selected Wikipedia pages and the annotated text accuracy was compared to the manual annotation. The system provided almost the same result as given by manual annotation.

Medelyan *et. al* [27] also used the same approach. The only significant difference detected was in the disambiguation phase. For the ambiguous terms, the authors used an approach by balancing between commonness and relatedness. Suppose a word has a few different senses. Commonness of each sense is measured by calculating the number of times that sense is used as a destination in Wikipedia. This is also known as prior probability. Relatedness of the sense

checks the surrounding text and detects the correct topic. Balancing these two parameters, commonness and relatedness, correct article on the web are linked for the correct sense.

In 2008, Milne and Witten [31] made an effort and showed how an automated process can be used to cross reference the documents with Wikipedia. Their approach also works in two steps, but in reverse order. The first step is disambiguation and the second step is link detection. Another key difference between their approach and Medelyan *et. al.* [27] approach is Milne and Witten used weighting context terms to balance between commonness and relatedness for the disambiguation. The authors have also explained how to link a phrase or even unstructured text to appropriate Wikipedia articles using machine learning techniques. For machine-learning, manually annotated Wikipedia articles and their links were used. Disambiguation and link detection steps were evaluated separately. To do this, around 700 articles from Wikipedia (version 2007) were selected as corpus. They made sure that each article had at least 50 links since these articles were used for training, configuration and evaluation. To evaluate the performance of each step, they used recall, precision and F-measure as metrics. For this, 50 randomly selected documents (250-300 words) were considered. These documents were gathered from The New York Times stories, and not Wikipedia, just to test whether the algorithm works on all documents or not. Short documents were selected just to help the evaluators to avoid excessive demand of their concentration region. Two tasks were performed during this experiment to evaluate the system's linking and disambiguation ability through human evaluation. The first task was focused on reviewing the quality of the links identified by the system, while the other one was focused on the identification of the links that should be identified by the algorithm but were missed. The results of both tasks were used to correct and update the original corpus of automatically-tagged articles and generate ground truth. The results showed that even for non-Wikipedia articles, high precision, recall and F-measure values can be obtained by using machine-learning technique.

In 2009, Gardner and Xiong [17] made an effort to better understand the automatic linking problem and defined the link detection problem as a sequence labelling problem. They concentrated on the link detection problem where conditional random fields (CRF) framework works as a probabilistic model. In this approach, all terms are given one of the three labels – O-

link (other link/no link), B-link (Begin link) or I-link (Intermediate link). CRF takes this sequential text/data and calculates the probability for each token based on the sequence. The tokens with maximum probabilities are considered as labels. The authors showed that almost perfect precision and high recall can be achieved by training the CRF using various types of features from the Wikipedia. For the evaluation purpose, for both - training and testing, Wikipedia 2008 dump was used. Around 500 articles were extracted from different categories including biology, business, health, language, mathematics and people. Short articles having less than four links - such as image and special pages were neglected in this dataset. Three metrics - precision, recall and F-measure - were used to check the performance of the classifier. They evaluated their system for all different categories. They also tested the system for the successful detection of B-link and I-link and the system showed almost 100% precision but received low recall value.

Kulkarni *et. al.* [24] proposed new algorithms by capturing an exchange between local spot-to-label potential and a global, document-level logic between entity labels. Here spot is used in reference to mention and label refers to the final article page label. Since previous works showed bias performances toward specific entity types like people and places, this effort was intended to provide a new approach that can give high recall value for the indexing and data mining purpose without reducing the precision value. For the evaluation purpose, the authors created a dictionary having entity IDs, their labels and mentions from Wikipedia 2008 dump. This included around 5.15 million entity IDs, including titles, redirections, disambiguation and category names. Unimportant entity IDs from the dump were removed by neglecting words composed purely of verbs, adverbs, conjunctions or prepositions and by not considering certain lexical patterns (e.g. fewer than three characters). As a ground truth, two datasets were used: one from Cucerzan's dataset (abbreviated as CZ dataset) which is publicly available. The issue with this dataset was it suffered from less number of annotations and was limited to few entity types. Therefore, the authors created a second dataset - IITB dataset. The IITB dataset was created using a browser-based annotation system. Popular sites' homepages were selected as documents for manual annotation. Around 19,000 annotations were collected by 6 volunteers. Both datasets – IITB and CZ - had high average ambiguity (5.3 and 18 per mention respectively). Hill climbing approach and LP relaxation approach were also implemented into their model and tested against CZ and

Milne-Witten (M&W) approaches. In this evaluation, recall, precision and F-measure were measured. In all the cases, new approach showed better performance than M&W and CZ approaches.

In 2010, Ferragina and Scaiella [15] introduced TagMe – an annotator that handles short text. It could be hard to annotate the ambiguous terms correctly for short texts since short texts like Tweets may not express the central topic. The authors claimed that TagMe was the first annotation system to handle short text fragments accurately. To achieve this annotation, two different phases are carried out. The first one is called anchor disambiguation and the other is anchor pruning (or link detection). The authors tested these two phases and also compared TagMe against two other annotator systems, namely M&W and Chakrabarti's systems, to show the accuracy of TagMe. The result shows that TagMe comes up with comparatively high precision and recall. For this evaluation, three datasets were used: WIKI DISAM 30, WIKI-ANNOT 30 and WIKI LONG. The first two datasets were having short texts and contained each fragment of around 30 words and the last one was used with long texts having at least 10 links. The goal was to measure the precision, recall and F measure of TagMe (for disambiguation and pruning phases separately) by comparing the real sense of the text and the result obtained from the experiment over those three datasets. In all cases, TagMe showed higher values of recall and F measure and hence, higher accuracy. Furthermore, a comparison experiment was performed on TagMe against the best available annotators – M&W and Chakrabarti's system - to check its accuracy over both short texts and long texts. TagMe performed better than other systems – especially for the short text.

In 2011, Mendes *et. al.* [29] introduced a new system called DBPedia Spotlight. Earlier, DBPedia was developed [2] as an interlink hub in the web of data that provides access to various data sources available in the Linked Open Data cloud. As a step forward, DBPedia Spotlight was introduced which enables the process to be automated to find and disambiguate accurate sense of the text from the DBPedia resources. As stated, DBPedia links are used for both DBPedia and DBPedia Spotlight, but the labels-titles are captured from Wikipedia article titles. DBPedia Spotlight also allows the users to configure the setting according to their requirements. For the evaluation of the DBPedia Spotlight, two different experiments were performed. First, the

disambiguation strategy of the DBPedia Spotlight was tested on 155,000 randomly selected wikilinks from the Wikipedia. The task was to measure the accuracy of the system to find the correct sense from various candidates based on the context. In the second experiment, DBPedia Spotlight system's performance was compared to other approaches – The Wiki Machine, Zemanta, Alchemy, Ontos and Open Calais. To do this, a set of manually annotated news articles – 35 paragraphs from New York Times having 8 different categories – were selected and compared with the gold standard. DBPedia Spotlight showed the competitive performance even for the different user specific parameters.

Hachey *et. al.* [19] analysed three different entity linking systems in their effort, but this effort was purely focused on the Named Entity Linking (NEL) process. NEL is the process to link popular people, location and organization names from the text to the web or Knowledge Base (KB). Just like other annotation systems, NEL also accomplishes the task in two basic steps: (1) search and (2) disambiguation. Hachey *et. al.* showed that the search phase contributes a major role in getting higher precision and recall and hence accuracy. The authors re-implemented three entity linking systems to see the relative importance of search phase and disambiguation phase. These three systems are described in [3] [8] [39]. The dataset, prepared earlier by National Institute of Standards and Technologies (NIST) for Knowledge Base Population task at TAC (Text Analysis Conference) for the year 2009 and 2010, were used. Actually, this dataset was created from a dump of Wikipedia 2008. As metrics, precision, recall and F-measure were used. This study showed that the search phase is highly important for any system's performance.

Ratinov *et. al.* [35] made an effort to take additional information from the Wikipedia link structure to improve the approach for the disambiguation to Wikipedia (D2W). The authors tried to utilize this additional information to find logical sets of disambiguation for a given document. Therefore, in this approach, all mentions in a document are disambiguated simultaneously to arrive at a coherent set of disambiguation. They called this approach as global approach and developed a new global D2W system called GLOW. The authors finally compared it with the traditional local approach - disambiguate each mention in a document separately, utilizing textual similarity between the document and each candidate disambiguation's Wikipedia page. For the evaluation purpose, four different datasets were used. Two of them were from previous

work and the other two were created by themselves. The first data set was from (Milne and Witten, 2008b) and it was a subset of the AQUAINT corpus of Newswire text which had similar structure to the hyperlink structure in Wikipedia. In this dataset, only "important" titles were considered. Repetitive titles, and titles that were considered uninteresting were not linked. The second data set was from (Cucerzan, 2007) and was taken from the MSNBC Newswire text. This dataset focused on disambiguating named entities after running NER (named entity reference) and co-reference resolution systems on Newswire text. All mentions of all the detected named entities were considered. Third dataset was a subset of ACE co-reference data set. This one had the advantage of having known mentions and their types and also co-references were resolved. Annotators from Amazon's Mechanical Turk were hired to link the first nominal mention of each co-reference chain to Wikipedia. The accuracy of these annotations having a majority of votes was around 85%. The rest of them were manually corrected to create ground truth for the experiment. The last dataset, Wiki, was the set of paragraphs from Wikipedia pages and mentions were from the existing hyperlinks in the Wikipedia text. 40 paragraphs were used for testing and the rest of them were used for training. In this evaluation, the disambiguation phase is divided and processed with two subsystems: Ranker and Linker. Ranker ranks the possible senses and linker links the most preferred sense to the article. Performance of ranker and linker were tested separately, both for Local and Global approaches, using all four datasets. Also the performance of the full system was also measured. Linker accuracy, ranker accuracy, recall, precision and F-measure were measured as metrics. As a result, in terms of ranker accuracy, global approach performed better than local approach. But for the linker accuracy, local approach gave better performance than global approach.

In 2012, Shen *et. al.* [38] proposed a new framework LINDEN for named entity linking. For the knowledge base, they selected YAGO – October 2010 version [43] [42] – which is an open domain ontology having massive amounts of entities and combines Wikipedia and WordNet [14]. LINDEN also works the same way in two steps: Mention recognition and Disambiguation. For the evaluation purpose, two datasets were used. (1) The news articles, used by Cucerzan dataset to test their system, were used as a first dataset since this dataset was publicly available [24]. The original dataset had 20 MSNBC news articles (2nd Jan, 2007), but when this dataset was downloaded for the experiment, one article was not available. Therefore, only 19 news

stories were considered. As a result, 614 entity mentions were obtained and 522 of them were linked manually to the knowledge base and remaining 92 mentions were unlinkable. (2) The second dataset was from TAC-KBP2009 (Text Analysis Conference) and KBP (Knowledge Base Population) having 3904 entity mentions. 1675 of them were mapped to their knowledge base and remaining 2229 mentions were unlinkable. Finally the performance of LINDEN was analysed and then compared with CZ and other systems. Various feature sets like linking probability, semantic associativity, semantic similarity, global coherence and their combinations were taken into consideration for the datasets and accuracy (measured as the total number of correct mentions divided by the total number of mentions) was measured and compared.

In 2013, Cornolti *et. al.* [6] felt that in the previous studies, the performances of semantic annotation systems were not measured on correct metrics. For example, suppose, the output mention given by the system is "Obama"; in the gold standard, it has been given as "Barak Obama" and suppose, the system predicts the associated entity as "President Obama". If the exact match is considered as a correct mention-entity, the above case would consider two wrong answers: (1) "Obama" should not be considered as a mention and (2) "Barak Obama" should be considered as a correct mention. But, one correct answer should be considered rather than two wrong answers to accurately measure the performance. Cornolti *et. al.* implemented a framework for benchmarking entity-annotation systems and evaluated different entity annotation systems. The main goal of the research was to develop this framework to evaluate system performance by involving publicly available datasets. So, to compare the systems fairly and accurately, the authors introduced (a) a hierarchy of entity-annotation problems and (b) a set of new measures to evaluate the performance of these systems. The experiments were carried out on all the publicly available systems and all publicly available datasets. The datasets used for this were AIDA/CoNLL, AQUAINT, MSNBC (the Newswire dataset), Meij (Tweets dataset) and IITB (web page). AIDA, Illinois Wikifier, DBPedia Spotlight, TagMe, Wikipedia Miner were the systems. As metrics, true-positive, true-negative, false-positive, false-negative, recall, precision and F-measures were used. In this evaluation work, TagMe showed better F-measure value on News dataset while Wikipedia Miner performed better on Tweets and web pages than other systems.

As described above, lots of efforts have been devoted to measuring the performance of various systems, in most of the efforts, the ground truths were created and recall, precision and F-measure or modified metrics were used. This has become a common approach. But in this research work, the semantic annotation systems are evaluated in a different way. The goal of this study is to include subjective opinions by including many human evaluators in the process of evaluation of the semantic annotators. Therefore, a systematic subjective evaluation method was undertaken to evaluate the semantic annotation systems. The detailed procedure has been covered in the next chapter. Here, a systematic approach is applied to perform this study by conducting a survey. Different types of texts have been processed on different publicly available systems and the outputs were sent to the participants. Likert type questionnaire was created and the responses from the participants were collected. By applying appropriate statistical tests, the performances of the systems have been evaluated.

2.3. Metrics

This section describes some of the core metrics that have been typically used to measure the performance of the semantic annotation system. As per the requirements, metrics were sometimes changed slightly in their respective evaluation efforts, but the basic definitions are described below. Also Table 2.1 shows the summary table for different metrics used in different evaluation works.

Precision: Precision is one of the measures to calculate the performance of an annotation system. This output is compared with the gold standard and the precision is measured. Precision is defined as a fraction of those given outputs that are correctly annotated by the system.

$$Precision = \frac{total\ no.of\ correct\ entites\ retrived\ by\ system}{total\ no.of\ retrived\ enties\ by\ system} \quad (2.1)$$

For example, given the gold standard contains *z* number of annotations, if the system gives *y* number of annotations and out of these y annotations, *x* annotations are correctly annotated by the system, then the system's precision is (*x/y*). The precision is measured in percentage.

Recall: Recall is also one of the most widely used measures to measure the performance of any annotation system. Recall is measured by dividing the number of correct annotations (given by the system) by total number of annotations that should be annotated (number of annotations in the gold standard).

$$Recall = \frac{total\ no.of\ correct\ entites\ retrived\ by\ system}{total\ no\ of\ entites\ in\ gold\ standard} \quad (2.2)$$

Again, for example, given the gold standard contains *z* number of annotations, if the system gives *y* number of annotations and out of these *y* annotations, *x* annotations are correctly annotated by the system, then the system's recall is *(x/z).*The difference between recall and precision is, precision measures the performance by counting the number of correct annotations in comparison to the number of output of the system while recall measures the performance by counting number of correct annotations in comparison to the number of total annotations in the gold standard.

F-measure: F-measure gives the combined effect of recall and precision. As mentioned in the example earlier, it is hard to say which system has better performance when system one has precision 100% but the recall is only 10% while system two gives precision 80% and recall 40%. F-measure has a formula that gives combined effect of recall and precision as follows:

$$Fmeasure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.3)$$

Ranker Accuracy: For any ambiguous terms, usually candidate generator generates possible candidates and gathers them in a candidate set. Ranker's duty is to rank them and pass this information to the Linker for further linking one of the candidates from this candidate set to the

ambiguous term. Ranker's accuracy is measured alone, irrespective of limitation of the candidate generator. The ranker's accuracy is measured based on the number of mentions for which the correct candidate is included in the given candidate set for the corresponding mentions. The correct candidate must fall within first *k* candidates in the generated candidate set, where *k* is a predetermined value.

$$Ranker\ Accuracy\ = \frac{No.of\ mentions\ having\ correct\ candidate\ in\ the\ candidate\ set\ returned\ by\ Ranker}{Total\ no.of\ mentions\ returned\ by\ Ranker} \quad (2.4)$$

Suppose candidate generator generates *x* number of candidates for any ambiguous term. A predefined number is used, let's say *k*. Ranker's duty is to rank those possible candidates from *1* to *k* and pass this detail to linker. If the correct candidate (the one that is in the gold standard) is ranked in between *1* to *k*, this means ranker is performing well and vice versa.

For better understanding, consider this example - "mercury" is an ambiguous term and the gold standard has a title for this is "mercury – the planet". Suppose candidate generator has generated 4 possible candidates: mercury (element), mercury (planet), mercury (mythology) and mercury (place). If the predefined value is 2 and suppose, ranker ranks the possible candidates as follows: (1) mercury (mythology) and (2) mercury (planet). Since the correct candidate – mercury (planet) - is in the outcome, the ranker has given us a correct result.

Linker Accuracy: Linker's duty is to find the correct candidate and provide the link. Continuing with the example above, if the linker chooses "mercury (mythology)" as a final candidate and links it with the mention, which means, the linker has failed to choose the correct candidate as the correct term in the gold standard is "mercury – the planet". As we can see, the linker's accuracy depends on the performance of the ranker. If ranker fails to give the correct candidate in first *k* candidates, there is no chance linker can find the correct candidate and link it.

Linker's performance is based on the linker's ability to find and link the correct candidate to the correct article.

$$Linker\ Accuracy\ = \frac{No.of\ mentions\ for\ which\ Linker\ links\ the\ correct\ candidate\ to\ the\ correct\ article}{Total\ no.of\ mentions} \quad (2.5)$$

Table 2.1 Metrics used in previous works

Papers	Metrics					
	Precision	**Recall**	**F-Measure**	**Linker's Accuracy**	**Ranker's Accuracy**	**Other**
Learning to link with Wikipedia [31].	✓	✓	✓	×	×	✓
DBPedia spotlight: Shedding light on the web of documents [29].	×	×	×	×	×	✓
Fast and accurate annotation of short texts with Wikipedia pages [15].	✓	✓	✓	×	×	×
Wikify! linking documents to encyclopedic knowledge [30].	✓	✓	✓	×	×	×
Local and global algorithms for disambiguation to Wikipedia [35].	✓	✓	✓	✓	✓	×
Collective annotation of Wikipedia entities in web text [24].	✓	✓	✓	×	×	×
Automatic link detection: a sequence labeling approach [17].	✓	✓	✓	×	×	×
Linden: linking named entities with knowledge base via semantic knowledge [38].	×	×	×	×	×	✓
LIEGE:: link entities in web lists with knowledge base [39].	×	×	×	×	×	✓
A Framework for Benchmarking Entity-Annotation Systems [6].	✓	✓	✓	×	×	✓

2.4. Corpus

In this section, some of the publicly available corpuses, used in some of the previous works, have been briefly described. Table 2.2 provides the list of corpuses used in previous works.

Wikipedia: The English version of Wikipedia, the biggest of all language editions, has about 10 million visits per hour and counts 4 million pages. It contains almost 30 million articles in about

287 different languages. Anyone having access to the site can edit the articles. The active editors (users that made at least 5 edits in a month) are 33,680 and the encyclopaedia is growing with 1,067 new articles per day. These manually edited articles are highly accurate [49]. Wikipedia dump of different versions were used in many evaluation works as corpora as well as training dataset. Also Wikipedia has been used as a base for number of natural language processing applications [3] [1] [16] [41].

Aquaint: Aquaint [31] consists of news articles in English made from three different sources: the Xinhua News Service, the New York Times News Service, and the Associated Press Worldstream News Service. Only important mentions are annotated in this data set and the structure of the dataset is almost same as the Wikipedia. Aquaint contains 50 documents with total of 727 annotated tags. Total 572 distinct topics are covered in this dataset. The average annotations per document are 14.5.

IITB: IITB [24] contains over a hundred of manually annotated texts drawn from popular Web pages about sport, entertainment, science and technology, and health. This dataset contains search engine queries. Popular sites' homepages are selected as documents for manual annotation. Around 19,000 annotations are collected by 6 volunteers. The number of distinct Wikipedia entities that are linked to are about 3,800. Among these, around 40% of the spots are labelled NA (No Attachment) and 60% spots are attached. Total number of documents is 107 in which total no of mentions found are 17,200. The average ambiguity per mention is 5.3.

MSNBC: MSNBC [15] dataset is taken from the MSNBC Newswire text. This dataset focuses on disambiguating named entities after running NER (named entity reference) and co-reference resolution systems on Newswire text. All mentions of all the detected named entities are considered. This dataset has only 20 documents. The total numbers of annotated tags are 658 in this dataset. Number of distinct topics covered in this dataset are 279. The average annotations per document are 32.9.

AIDA/CoNLL: This dataset was built on CoNLL – Conference of Natural Language Learning - 2003 entity recognition task [21]. The data were taken from Reuters Corpus V1 news. A large amount of mentions were annotated but the mentions having common names were not annotated. Each occurrence of a mention was annotated. In total this dataset contains 34,956 mentions.

Many different types of topics are covered in the dataset having total of 1,393 articles. It has average of 25 mentions per articles.

Cucerzan dataset: Cucerzan dataset, also known as "CZ dataset", was taken from MSNBC newswire text in 2007 [8]. This dataset focused on disambiguating named entities after running NER (named entity reference) and co-reference resolution systems on newswire text. All mentions of all the detected named entities were considered. This dataset is publicly available but the disadvantage is that the annotations are sparse and limited to a few entity types only (mostly person and place names).

Meij: In this dataset, tweets were annotated [28]. Meij dataset was introduced in one of the evaluation works where the goal of the study was to find concept from tweets having total characters less than 140. The dataset was created to compare different systems' performances for the short texts. To prepare for the experiment, around 500 tweets were collected with maximum of 50 retrieved concepts. Each tweet text was tokenized where punctuation and capitalization were removed. Mentions, URL and Symbols were also removed.

Table 2.2 Datasets used in previous works

Papers	Wikipedia	CZ-data	AIDA/ CONLL	Aquaint	MSNBC	TAC	ACE	IITB	Meij
Learning to link with Wikipedia [31].	✓	×	×	✓	×	×	×	×	×
Fast and accurate annotation of short texts with Wikipedia pages [15].	✓	×	×	×	×	✓	×	×	×
Wikify! linking documents to encyclopedic knowledge [30].	✓	×	×	×	×	×	×	×	×
Local and global algorithms for disambiguation to Wikipedia [35].	✓	×	×	✓	✓	×	✓	×	×

Collective annotation of Wikipedia entities in web text [24].	✓	×	×	×	×	×	×	✓	×
Automatic link detection: a sequence labeling approach [17].	✓	×	×	×	×	×	×	×	×
Linden: linking named entities with knowledge base via semantic knowledge [38].	✓	✓	×	×	×	✓	×	×	×
LIEGE: link entities in web lists with knowledge base [39].	✓	×	×	×	×	×	×	×	×
A Framework for Benchmarking Entity-Annotation Systems [6][7].	×	×	✓	✓	✓	×	×	✓	✓
No. of times the dataset is included in evaluation work	8	1	1	3	2	2	1	2	1

TAC: In 2009 TAC – Text Analysis Conference, National Institute of Standards and Technologies (NIST) introduced manually annotated named entity linking data [19]. This dataset was the first large TAC knowledge base, which was extracted from a dump of Wikipedia in, 2008. The TAC dataset collection mainly contains newswire and blogs. Each knowledge base node contains the Wikipedia article title, Wikipedia article text, a predicted entity type and a key value list of information extracted from the article's infobox. The TAC dataset contains various nodes including 200,000 person nodes, 60,000 org nodes and 300,000 a variety of non-entity nodes.

As one can see in Table 2.2, Wikipedia has been used most of the times since Wikipedia is an encyclopaedia having mentions annotated by its users – human annotators. Also Wikipedia is available in 287 languages and it is free.

2.5 Systems

Many semantic annotation systems have been developed so far. Some of them are briefly described below. They are: (1) DBPedia Spotlight, (2) TagMe, (3) Wikipedia Miner, (4) Wikimeta, (5) Illinois Wikifier and (6) Denote.

DBPedia Spotlight: DBPedia Spotlight [9] extracts information from Wikipedia and uses this information as its knowledge base. DBPedia Spotlight uses multi domain ontology. Currently, DBPedia Spotlight supports 10 languages, but English has the largest version compare to other languages. The English version of DBPedia knowledge base has 3.5 million entities, including person names, places, organizations, creative works, species and diseases. DBPedia Spotlight application uses DBPedia terms for annotating the web data. DBPedia Spotlight detects mentions and concepts from DBPedia resources. DBPedia Spotlight identifies mentions from text and matches with that entity in DBPedia dataset.

TagMe: TagMe is a system that can identify spots from given text and link them to relevant the Wikipedia pages in a fast and effective way. TagMe [45] gives better result on short text compared to long text. The first version of TagMe was introduced in 2010 and later, the flexibility, precision and speed were improved in Aug 2012. TagMe has different services such as tagging, spotting and relating. Tagging is the main TagMe service that is used to annotate a text. In the process of tagging, all annotations found in the input text are considered. An attribute is linked to each annotation. This attributes estimates the relatedness of the annotation with respect of to the input text. A predetermined value (threshold value) of attribute is selected below which the annotations having lesser value are discarded. In the spotting service, the spots are identified in a text. The only difference is, there is no need to disambiguate the annotation and no links to Wikipedia pages are provided. So in this service TagMe identifies relevant spots in the text only. The importance of each spot is measured in the link probability and that can be used to improve the result. The relatedness between two topics is computed using the relating service. Relatedness is a value that defines how much two topics are semantically similar to each other. This value ranges in the interval [0, 1]. The disambiguation is done with a voting scheme where the concept having the highest value of relatedness is selected and tied to the mention. Currently, TagMe is available in two languages: English and Italian.

Wikipedia Miner: Wikipedia Miner [51] was implemented from the Wikification algorithm. In this system, disambiguation is done before the identification of mentions. The disambiguation phase is performed using a machine learning approach. In this approach, training is provided using links taken from Wikipedia pages. In other words, Wikipedia data is not used for linking to the articles only, but also as training data to improve efficiency. For the ambiguous terms, the senses of each term are compared against each other to find the most relevant candidate. Wikipedia Miner computes relatedness by measuring connections between terms, articles and IDs.

Wikimeta: Wikimeta [50] is used to annotate the text for text mining and data analysis. It also provides the links to the Wikipedia articles. Extra information is also provided along with the annotated texts and is connected to the documents of Wikimeta itself. Wikimeta performs better with named entity labelling. Currently, Wikimeta is available in three languages: English, French and Spanish. Separate models are used for each language.

Illinois Wikifier: Illinois Wikifier [22], usually known as Wikifier, also provides links to Wikipedia. Same as Wikipedia Miner, it also uses Wikification step. Here the disambiguation step is considered as an optimization problem. This system tries to disambiguate all terms together unlike most other system that does disambiguate for each mention separately. Currently, Illinois Wikifier is available in two languages: Italian and English.

Denote: Denote [10] is one of the semantic annotation systems that can extract concepts and keywords from the given documents. While annotating documents, the system gives annotations and at the same time, it also gives related contents of those documents. Related contents help in giving more knowledge of that document. Annotations of this system give images of those annotations. These contents of the document make the topic more clearer based on images. Related contents include images of content, labels, categories of the label, and confidence scores. The system has customizable settings. With the use of these settings, users can change their preferences accordingly.

Next chapter describes the detailed methodology that has been followed to evaluate these systems.

CHAPTER 3

METHODOLOGY

This chapter provides the detailed procedure that has been followed to perform the systematic evaluation of different systems using our proposed method. It describes the questionnaire developed for the subjective evaluation, the corpus details and the systems as well.

3.1 Questionnaire

For the subjective evaluation, a questionnaire is required asking multiple choice type questions. The questions need to be focusing on various capabilities on the semantic annotation systems. The abilities, such as disambiguating various terms, disambiguating named entities, searching required keywords, linking mentions to the correct articles on the Web, number of annotations provided, recognizing the central theme etc., need to be covered. Questions are also required to collect the participants' feeling about how the system performed. The questions should be non-repetitive and also specific to only one capability per question. To accommodate the overall performance feedback, a question is required to decide whether the participant could determine the source of the annotation (human or machine). Responses to this question could indicate the overall performance of the system.

Hence, a questionnaire with a total 13 different questions was prepared for the participants. The idea was to test the systems on three types of texts: Long Text, Short Text and Tweets. So, Question 1 is about the length of text that is being evaluated. The remaining questions are organized in 4 different categories: (1) Questions 2-5 are associated to spotting and disambiguation ability of the system, (2) Questions 6-8 are related to usefulness of the system, (3) Questions 9-12 are related to comprehensiveness and finally, (4) Question 13 is for the overall feedback. Questions 2-9 are Likert type questions with five options: Strongly Agree, Agree, Neutral, Disagree and Strongly Disagree. Questions 10-13 are also Likert type questions with different options mentioned later in this chapter during their analyses.

As stated earlier, question 1 is created just to store the data in the right category for the analysis. This question is mandatory. So during the analysis stage, the type of text category can be classified and organized easily.

1. The word document you read is: Is it Long Text, Short Text or Tweet?

Questions 2-5 mainly emphasize the system's capability to perform two main tasks: (1) to find the correct mentions and (2) to disambiguate the terms. Question 2 collects the answers about whether the system is able to identify the correct sense of the ambiguous terms. Question 3 asks the user if the system can provide accurate links for the generated spots, while question 4 takes care of the relevancy – whether given articles are relevant to the text. Question 5 gets the answer on whether the system is able to annotate important keywords that should be annotated.

2. The annotator was able to correctly find the best sense for the annotated phrases.

3. The highlighted texts and their related annotations are accurate.

4. The highlighted texts and their related annotations are relevant.

5. The annotator was able to identify the main phrases that needed to be annotated.

Questions 6-8 are related to the system's ability to assist the user to better understand the given document. Question 6 and 7 ask whether the system is able to provide more information and appropriate links by annotating useful keywords. Question 8 checks if the system is capable to locate the main topic in the given document.

6. The annotations helped me understand the document in ways which would not be otherwise possible.

7. The annotations were quite informative beyond what was understood from the document alone.

8. The annotator has been able to identify the central theme of the document correctly.

Questions 9-12 focus on the user's feedback about the completeness of the system. Question 9 is also a Likert type question with five options and mainly focuses on system's ability to accurately recognize the named entities. Questions 10-12 are aiming for the quality and quantity of annotations the system has provided. Some annotation systems annotate the same mention multiple times. These questions also check the users' feeling about whether they like this or not.

9. The annotator was able to identify famous people, places and organizations.

10. The number of annotations provided for this document by the annotator system was in enough quantity.

11. The annotations produced by the annotation tool are relevant to the topic.

12. The annotator recognized and annotated the entities that I would select and annotate myself.

Finally, question 13 asks the user, his/her belief whether the document is annotated by human annotator or machine. This shows the overall performance of the system. If the users cannot identify that the document is annotated by the computer clearly, then one can say that the system has performed well and provided the annotation that usually human annotator can annotate.

13. What do you believe the source of the annotations is?

3.2 Corpus for the Subjective Evaluation

The goal of this study is to perform a systematic subjective evaluation of semantic annotators. Since many human participants are involved, different people may have different interests and knowledge. Therefore, six different categories/topics are involved. These categories are: Entertainment, Advice, Sports, Disease, Technology and Others. Also each category has two types of texts: Formal texts and Informal texts. News articles, published articles are considered as formal texts, while comments and blogs, having some slang language, can be considered as informal text. Formal texts have been included from different published articles from the web. Also some of the formal texts are collected from three of the publicly available datasets explained in Chapter 2. They are: Aquaint, IITB and MSNBC. The informal texts are included from the comments and blogs. Previous studies show that different systems perform differently depending on the length of the document given. So different types of texts are also one of the requirements to see how different systems behave for the given type of texts. For this reason, the documents are further divided into two types of texts: (1) Long Texts (texts having words, roughly more than half page), (2) Short Texts (texts with words less than half page). Additionally, tweets from the social websites are collected as a third type of text. So, total 60 documents are gathered. Out of these 60 documents, 22 documents are of long texts, 22

documents are of short texts and remaining 16 are tweets. All these documents are in English language only. Figure 3.1 shows the structure of the corpus created for this subjective evaluation.

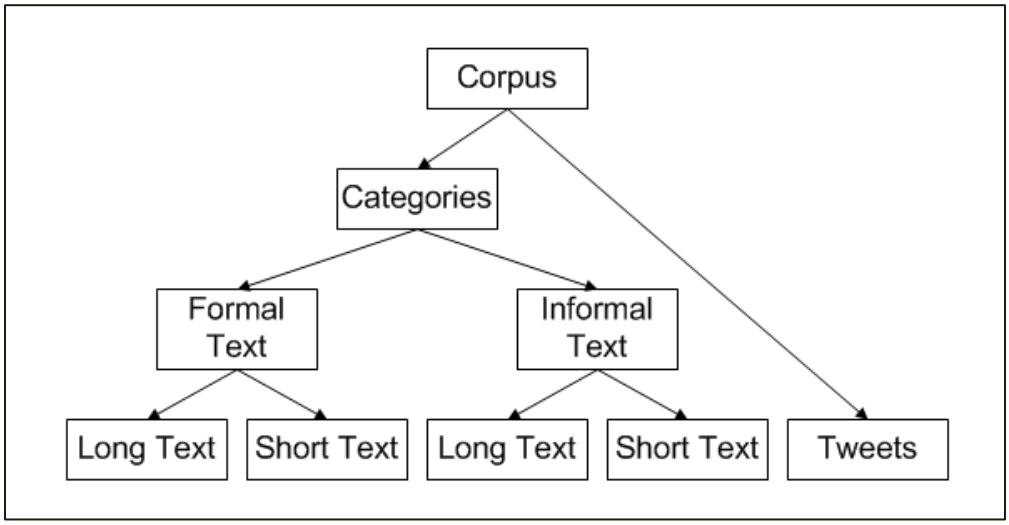

Figure 3.1 Structure of the Corpus

3.3 Systems

Six different semantic annotation systems have been evaluated here. They are: (1) DBPedia Spotlight, (2) TagMe, (3) Wikipedia Miner, (4) Wikimeta, (5) Illinois Wikifier and (6) Denote. The demos for all these systems are publicly available. Denote and DBPedia Spotlight provide links to DBPedia online database (http://DBPedia.org) while TagMe, Wikipedia Miner, Wikimeta and Illinois Wikifier point to Wikipedia (http://www.wikipedia.org).

3.4 Distribution and Response Collection

All the 60 documents were processed using each system. Most of these systems offer customizable settings and user can annotate more or less links. But here, the documents were annotated using the default settings offered by the respective systems. The generated output were copied and pasted into the word documents. The word documents provide hyperlinks with different colour, which were easy to open. So it was convenient for the participants to visualize the output. So total of 360 unique system-document pairs were created. These system-document pairs were randomized and distributed among the participants in such a way that one participant

did not get the same text type and/or system twice. The participants were not informed which system output they had received (complete anonymization). So the system names were kept unknown to avoid any potential bias. 60 participants' email addresses were collected. The participants were informed about the semantic annotation systems, how they perform and the goal for this study. This was done before they had received any survey links. Since this study required human participation, a consent form, mentioning that they were not forced to finish this survey and could skip any question or stop the survey at their wish, was also sent. Randomly selected documents were emailed to the participants along with the survey links. The participants were asked to read the given document first and then fill the survey accordingly. At last, the responses were collected in Google drive in Ryerson secure server once the participants submitted their answers. Responses for different systems were collected in six different spread sheets and as described earlier, Question 1 was asked to distinguish the text type. However, the participants were instructed again about the definition of long text, short text and tweet during this question. Therefore, it has been assumed that the participants correctly selected the document type i.e. long text, short text or tweet.

Finally, the analysis was performed on each question. Each question was analyzed using standard statistical tests. Chapter 4 provides the detailed analysis on the collected responses.

CHAPTER 4

ANALYSES

This chapter gives the detailed analyses of the responses collected from the participants. Questions 2-9 were Likert type questions and the first sub section analyses them. Questions 10-12 could also be treated as Likert type questions and their analyses are described in the second sub section. Finally, the responses for the overall feedback (i.e. question 13) are analysed in the third sub section.

4.1 Analyses on Question 2-9

In our questionnaire, Likert type questions have one of the following answers: (1) Strongly Agree, (2) Agree, (3) Neutral, (4) Disagree and (5) Strongly Disagree. Here the difference between "Strongly Agree" to "Agree" and "Agree" to "Neutral" is not equal. The same is true for all five options. Because of this, these data cannot be considered as interval data, but have to be treated as ordinal data.

In this survey, six different systems (DBPedia Spotlight, Illinois Wikifier, Wikipedia Miner, Denote, Wikimeta and TagMe) and three types of texts (long text, short text and tweets) are present. The responses from the participants were used for the analysis. Here the responses gathered are independent since no participant has given response for the same "system-text type" combination more than once.

So the analysis is performed considering two cases as follows:

Case 1: Type of text will be the independent variable for any given system and the responses the participants provided will be treated as the dependent variable.

Case 2: Six systems will be the independent variable for any given type of text and the responses the participants provided will be treated as dependent variable.

The analysis is carried out for each question separately and for both cases.

Now, as stated earlier, Questions 2-9 are Likert type questions and need to be treated as ordinal data. In statistics, different analysis methods are available to analyse ordinal data. One of them is Kruskal-Wallis Test (KW Test). KW Test is useful to determine whether any significant difference has been observed or not. Another test is the Mann-Whitney U Test which also gives the same information. The major difference between these two tests is KW Test is applied when more than two independent variable groups are present where the Mann-Whitney U Test is applied to test only two groups. In this analysis, Case 1 has three types of texts as independent variables and Case 2 has six systems as independent variables. Therefore, as a first step, KW Test is performed on data to find out if there any significant difference exists for any particular independent variable. If there is any, as the second step, U Test is performed to find out which variable is showing the significant difference.

To better clarify, we will provide the details of the analysis for Question 2. The same process is repeated for Questions 3-9 of the questionnaire. Table 4.1 shows the values assigned to the responses for the calculation.

Table 4.1 Values assigned to the options for Questions 2-9

Response	Value assigned
Strongly Agree	5
Agree	4
Neutral	3
Disagree	2
Strongly Disagree	1

4.1.1 Question 2 - Case 1

Independent variable: Types of texts

Dependent variables: Responses from the participants for a given system

Step 1: Kruskal Wallis Test

In this test, we will try to find if any type of text causes significantly different performance for given system. For this, consider following two hypotheses:

Null Hypothesis: The given system does not show significant difference in the performance for any type of texts.

Alternative Hypothesis: The given system shows significant difference in the performance for any type of texts.

The data was organized and processed in SPSS tool [40]. Figure 4.1 is a snapshot from the SPSS tool and shows how the data was organized for this step.

Type of Text	Dbp.Spotlight	TagMe	Wikipediaminer	Wikimeta	II.Wikifier	Denote
Long Text	Strongly Agree	Strongly Agree	Agree	Neutral	Neutral	Agree
Long Text	Strongly Agree	Agree	Strongly Agree	Agree	Strongly Agree	Agree
Long Text	Agree	Agree	Disagree	Strongly Agree	Agree	Disagree
Long Text	Strongly Agree	Agree	Strongly Agree	Disagree	Disagree	Neutral
Long Text	Neutral	Disagree	Strongly Agree	Strongly Agree	Strongly Agree	Strongly Agree
Short Text	Agree	Neutral	Agree	Strongly Agree	Strongly Agree	Disagree
Short Text	Agree	Strongly Agree	Neutral	Disagree	Agree	Agree
Short Text	Disagree	Strongly Agree	Neutral	Agree	Agree	Neutral
Short Text	Neutral	Strongly Agree	Agree	Agree	Agree	Agree
Short Text	Agree	Agree	Agree	Agree	Neutral	
Short Text	Neutral		Strongly Agree			
Short Text	Agree		Strongly Agree			
Short Text	Agree		Agree			
Tweets	Agree	Agree	Strongly Disagree	Disagree	Strongly Agree	Strongly Disagree
Tweets	Disagree	Disagree	Disagree		Strongly Agree	Agree
Tweets	Strongly Agree	Strongly Agree	Disagree	Neutral	Agree	Strongly Disagree
Tweets	Agree	Strongly Agree	Disagree	Strongly Agree	Agree	Agree

Figure 4.1 Snapshot of SPSS tool for Case 1- KW Test

Output:

After performing the KW Test, the following output is produced. Figure 4.2 is a snapshot of SPSS output.

Test Statistics[a,b]

	Dbp.Spotlight	TagMe	Wikipediaminer	Wikimeta	II.Wikifier	Denote
Chi-Square	4.999	5.604	15.509	1.709	3.791	1.725
df	2	2	2	2	2	2
Asymp. Sig.	.082	.061	.000	.426	.150	.422

a. Kruskal Wallis Test

b. Grouping Variable: Type_text

Figure 4.2 Output of SPSS tool for Case 1 – KW Test

Interpretation:

Here the threshold value considered is 0.05 (95% confidence level) which means the Null hypothesis will be rejected for the variable having *p-value* (Asymp. Sig. in Figure 4.2) less than 0.05. We fail to reject the Null hypothesis when the *p*-value is greater than or equal to 0.05. In other words, any variable having *p-value* less than 0.05 shows significant difference in performance.

As we can see in the output from KW Test, for Wikipedia Miner, *p-value* is equal to 0.000 which is less than the threshold value. That means, one of the text types is causing Wikipedia Miner to show significantly different performance. Now the second step, the Mann-Whitney U Test, is carried out to determine the type of text that is causing the significant difference in the performance for Wikipedia Miner. As stated earlier, U Test can be performed only on two groups. As we have three types of texts, total three combinations of two groups are possible: (a) long texts and short texts, (b) short texts and tweets and (c) long texts and tweets. The following step shows the Mann-Whitney U Test for each situation.

Step 2: Mann Whitney U Test

In this test, we will try to find which type of text causes Wikipedia Miner to perform significantly different. Figures 4.3, 4.6 and 4.9 show how the data were organized in SPSS tool for situation *a*, *b* and *c* respectively.

Situation (a): *System: Wikipedia Miner*

Grouping variable: Long texts vs. Short texts

Type of Text	Wikipediaminer
Long Text	Agree
Long Text	Strongly Agree
Long Text	Agree
Long Text	Agree
Long Text	Strongly Agree
Long Text	Agree
Short Text	Neutral
Short Text	Neutral
Short Text	Agree
Short Text	Agree
Short Text	Strongly Agree
Short Text	Strongly Agree

Figure 4.3 Snapshot of SPSS tool for Case 1- U Test (Situation (*a*))

Output:

After performing the U Test, following output is produced. Figure 4.4 and 4.5 are the snapshot of SPSS output.

Ranks

	System	N	Mean Rank	Sum of Ranks
Wikipediaminer	Long Text	16	20.72	331.50
	Short Text	20	16.73	334.50
	Total	36		

Figure 4.4 Output of SPSS tool - Rank Table for Case 1- U Test (Situation (*a*))

Test Statistics[b]

	Wikipediaminer
Mann-Whitney U	124.500
Wilcoxon W	334.500
Z	-1.251
Asymp. Sig. (2-tailed)	.211
Exact Sig. [2*(1-tailed Sig.)]	.262[a]

Figure 4.5 Output of SPSS tool - *p-value* for Case 1- U Test (Situation (*a*))

Here, the *p-value* in Figure 4.5 is greater than 0.05, which means Wikipedia Miner shows no significant difference in performance if the type of text is either long text or short text.

Situation (b): *System: Wikipedia Miner*

Grouping variable: Short texts vs. Tweets

Type of Text	Wikipediaminer
Short Text	Strongly Agree
Short Text	Neutral
Short Text	Neutral
Short Text	Agree
Short Text	Agree
Short Text	Strongly Agree
Short Text	Strongly Agree
Short Text	Agree
Tweets	Strongly Disagree
Tweets	Disagree
Tweets	Disagree
Tweets	Disagree
Tweets	Disagree

Figure 4.6 Snapshot of SPSS tool for Case 1- U Test (Situation (*b*))

Output:

After performing the U Test, following output is produced. Figure 4.7 and 4.8 are the snapshot of SPSS output.

Ranks

	System	N	Mean Rank	Sum of Ranks
Wikipediaminer	Short Text	20	22.53	450.50
	Tweets	15	11.97	179.50
	Total	35		

Figure 4.7 Output of SPSS tool - Rank Table for Case 1- U Test (Situation (*a*))

Test Statistics[b]

	Wikipediaminer
Mann-Whitney U	59.500
Wilcoxon W	179.500
Z	-3.196
Asymp. Sig. (2-tailed)	.001
Exact Sig. [2*(1-tailed Sig.)]	.002[a]

Figure 4.8 Output of SPSS tool - *p-value* for Case 1- U Test (Situation (*b*))

Here the *p-value* in Figure 4.8 is less than 0.05, which means Wikipedia Miner shows significant difference in the performance. To decide for which type of text, we check the rank value in Figure 4.7- rank table. Mean rank for short text and tweets are 22.53 and 11.97 respectively, which means Wikipedia Miner shows significantly better performance when the given text type is short text compared to tweets.

Situation (c): *System: Wikipedia Miner*

Grouping variable: Long texts vs. Tweets

Type of Text	Wikipediaminer
Long Text	Agree
Long Text	Agree
Long Text	Strongly Agree
Long Text	Strongly Agree
Long Text	Agree
Long Text	Strongly Agree
Long Text	Disagree
Tweets	Strongly Disagree
Tweets	Disagree
Tweets	Disagree

Figure 4.9 Snapshot of SPSS tool for Case 1- U Test (Situation (*c*))

Output:

After performing the U Test, the following output is produced. Figure 4.10 and 4.11 are the snapshots of SPSS output.

Ranks

	System	N	Mean Rank	Sum of Ranks
Wikipediaminer	Long Text	16	21.16	338.50
	Tweets	15	10.50	157.50
	Total	31		

Figure 4.10 Output of SPSS tool - Rank Table for Case 1- U Test (Situation (*c*))

Test Statistics[b]

	Wikipediaminer
Mann-Whitney U	37.500
Wilcoxon W	157.500
Z	-3.426
Asymp. Sig. (2-tailed)	.001
Exact Sig. [2*(1-tailed Sig.)]	.001[a]

Figure 4.11 Output of SPSS tool *p-value* for Case 1- U Test (Situation (*c*))

Again, the *p-value* in Figure 4.10 is less than 0.05, which means Wikipedia Miner shows substantial difference in performance. From Figure 4.11 - rank table, the mean rank for long text is 21.16 and for tweets, it is 10.50, which means Wikipedia Miner shows significantly better performance when the given text type is a long text compared to tweets.

Conclusion for Q2 - Case 1: By performing KW test and U test, one can say that Wikipedia Miner shows significantly better performance for long texts and short texts compared to tweets regarding its ability to correctly find the best sense for the annotated phrases. The remaining systems, for example – Wikimeta perform almost same regardless of the given type of text.

4.1.2 Question 2 - Case 2

Independent variables: Semantic Annotation Systems

Dependent variables: Responses from the participants for a given type of text

Step 1: Kruskal Wallis Test

In this test, we will try to find whether any annotation system performs significantly different for the given type of text.

Null Hypothesis: For the given type of text, the systems do not show significant difference in the performance.

Alternative Hypothesis: For the given type of text, the systems show significant difference in the performance.

Figure 4.12 is a snapshot of the SPSS tool and shows how the data was organized for this step.

System	Long Text	Short Text	Tweets
Dbp.Spotlight	Agree	Agree	Agree
Dbp.Spotlight	Agree	Strongly Agree	Disagree
Dbp.Spotlight	Agree	Agree	Strongly Agree
TagMe	Neutral	Agree	Agree
TagMe	Neutral	Agree	Disagree
TagMe	Agree	Strongly Agree	Strongly Agree
Wikipediaminer	Strongly Agree	Strongly Agree	Disagree
Wikipediaminer	Strongly Agree	Neutral	Agree
Wikipediaminer	Agree	Agree	Agree
Wikimeta	Agree	Disagree	Strongly Disagree
Wikimeta	Disagree	Strongly Disagree	Neutral
Wikimeta	Disagree	Agree	Disagree
II.Wikifier	Disagree	Agree	Agree
II.Wikifier	Agree	Agree	Agree
II.Wikifier	Agree	Strongly Agree	Agree
Denote	Agree	Neutral	Neutral
Denote	Strongly Agree	Disagree	Agree
Denote	Neutral	Agree	Strongly Disagree

Figure 4.12 Snapshot of SPSS tool for Case 2 - KW Test

Output:

After performing the KW Test, the following output is achieved. Figure 4.13 is a snapshot of SPSS output.

Test Statistics[a,b]

	Long Text	Short Text	Tweets
Chi-Square	13.512	11.151	19.669
df	5	5	5
Asymp. Sig.	.019	.048	.001

a. Kruskal Wallis Test

b. Grouping Variable: System

Figure 4.13 Output of SPSS tool for Case 2 – KW Test

Interpretation:

As we can see in the output from Figure 4.13 - KW Test, all three types of texts shows *p-value* less than 0.05. That means, U Test is required to find which systems show noteworthy difference for the given type of text. We have six systems and three types of texts, so we need to perform U Test 15 times $\left(\frac{6!}{(2!)(6-2)!} = \frac{6\times5\times4\times3\times2\times1}{2\times1\times4\times3\times2\times1} = 15\right)$ for each type of text, so total 45 times.

Just for explanation purposes, out of 45 combinations, only one combination has been shown below.

Step 2: Mann Whitney U Test

Situation (a): *Type of text: Long text*

Grouping variables: DBPedia Spotlight vs. Wikipedia Miner

In this test, we will try to find which system, between DBPedia Spotlight and Wikipedia Miner, performs significantly better if the given type of text is long text. Figure 4.14 shows how the data was passed in to SPSS to perform this step.

System	Long Text
Dbp.Spotlight	Agree
Dbp.Spotlight	Agree
Dbp.Spotlight	Agree
Dbp.Spotlight	Agree
Dbp.Spotlight	Agree
Dbp.Spotlight	Agree
Dbp.Spotlight	Neutral
Dbp.Spotlight	Disagree
Dbp.Spotlight	Disagree
Wikipediaminer	Agree
Wikipediaminer	Strongly Agree
Wikipediaminer	Agree

Figure 4.14 Snapshot of SPSS tool for Case 2- U Test (Situation (*a*))

Output:

After performing the U Test, the following output is achieved. Figure 4.15 and 4.16 are the snapshots of SPSS output.

Ranks

	System	N	Mean Rank	Sum of Ranks
Long Text	Dbp.Spotlight	21	15.69	329.50
	Wikipediaminer	16	23.34	373.50
	Total	37		

Figure 4.15 Output of SPSS tool - Rank Table for Case 2- U Test(Situation (*a*))

Test Statistics[a]

	Long Text
Mann-Whitney U	98.500
Wilcoxon W	329.500
Z	-2.288
Asymp. Sig. (2-tailed)	.022
Exact Sig. [2*(1-tailed Sig.)]	.032[a]

Figure 4.16 Output of SPSS tool - *p-value* for Case 2 - U Test (Situation (*a*))

Here the *p-value* is less than 0.05, which means for the long text, either Wikipedia Miner or DBPedia Spotlight is showing significantly better performance. The mean ranks for DBPedia Spotlight and Wikipedia Miner are 15.69 and 23.34 respectively, which means Wikipedia Miner shows significantly better performance than DBPedia Spotlight when the given type of text is long text.

The following tables – Table 4.2, 4.3 and 4.4 show the summary of Mann Whitney U Test Output for Long Text, Short Text and Tweets.

Summary Table for Q2 – Case 2 – Step 2 – Long Texts

Table 4.2 Output of SPSS tool - p-value and Ranks for Question 2 - Case 2 - U Test (Long Text)

Long					
Systems pairs	**Asymp. Sig. (*p-value*)**	**System**	**N**	**Mean**	**Sum Of**
Dbp.Spotlight_Vs._TagMe	0.749				
Dbp.Spotlight_Vs._Wikipedia	0.022	Dbp.Spotlight	21	15.69	329.5
		Wikipedia	16	23.34	373.5
Dbp.Spotlight_Vs._Wikimeta	0.357				
Dbp.Spotlight_Vs._Il.wikifier	0.105				
Dbp.Spotlight_Vs._Denote	0.749				
TagMe_Vs._Wikipedia Miner	0.024	TagMe	20	15.18	303.5
		Wikipedia	16	22.66	
TagMe_Vs._Wikimeta	0.433				
TagMe_Vs._Il.Wikifier	0.103				

TagMe_Vs._Denote	0.826				
Wikipedia Miner_Vs._Wikimeta	0.004	Wikipedia	16	26.03	416.5
		Wikimeta	23	15.8	363.5
Wikipedia Miner_Vs._Il.Wikifier	0.451				
Wikipedia Miner_Vs._Denote	0.016	Wikipedia	16	23.63	378
		Denote	21	15.48	325
Wikimeta_Vs._Il.Wikifier	0.18				
Wikimeta_Vs._Denote	0.578				
Il.Wikifier_Vs._Denote	0.7				

As we can see in Table 4.2, for the long text, the *p-values* (Asymp. Sig.) for the combinations – "DBPedia Spotlight – Wikipedia Miner", "TagMe – Wikipedia Miner", "Wikipedia Miner - Wikimeta" and "Wikipedia Miner - Denote" are less than 0.05. So by checking the mean rank values which are higher for Wikipedia Miner, one can conclude that, "If the same long text is given to the above six systems, Wikipedia Miner will perform significantly better than TagMe, DBPedia Spotlight, Wikimeta and Denote".

Summary Table for Q2 – Case 2 – Step 2 – Short Texts

Table 4.3 Output of SPSS tool - *p-value* and Ranks for Question 2 - Case 2 - U Test (Short Text)

Short Text					
Systems pairs	**Asymp. Sig (*p-value*)**	**System**	**N**	**Mean**	**Sum Of**
Dbp.Spotlight_Vs._TagMe	0.154				
Dbp.Spotlight_Vs._Wikipedia	0.072				
Dbp.Spotlight_Vs._Wikimeta	0.304				
Dbp.Spotlight_Vs._Il.wikifier	0.716				
Dbp.Spotlight_Vs._Denote	0.973				
TagMe_Vs._Wikipedia Miner	0.895				
TagMe_Vs._Wikimeta	0.026	TagMe	1	21.15	359.5
		Wikimeta	1	13.85	235.5
TagMe_Vs._Il.Wikifier	0.208				
TagMe_Vs._Denote	0.146				
Wikipedia Miner_Vs._Wikimeta	0.007	Wikipedia	2	23.18	463.5
		Wikimeta	1	14.09	239.5
Wikipedia Miner_Vs._Il.Wikifier	0.91				
Wikipedia Miner_Vs._Denote	0.063				
Wikimeta_Vs._Il.Wikifier	0.91				
Wikimeta_Vs._Denote	0.281				
Il.Wikifier_Vs._Denote	0.666				

Same way, from Table 4.3, one can conclude that for a given short text, TagMe and Wikipedia Miner will perform significantly better than Wikimeta.

Summary Table for Q2 – Case 2 – Step 2 – Tweets

Table 4.4 Output of SPSS tool - *p-value* and Ranks for Question 2 - Case 2 - U Test (Tweets)

Tweets					
Systems pairs	**Asymp. Sig(p-value)**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._TagMe	0.002	Dbp.Spotlight	13	8.62	112
		TagMe	11	17.09	188
Dbp.Spotlight_Vs._Wikipedia Miner	0.792				
Dbp.Spotlight_Vs._Wikimeta	0.975				
Dbp.Spotlight_Vs._Il.wikifier	0.007	Dbp.Spotlight	13	9.54	124
		Il.wikifier	13	17.46	227
Dbp.Spotlight_Vs._Denote	0.802				
TagMe_Vs._Wikipedia Miner	0.001	TagMe	11	18.95	208.5
		Wikipedia Miner	15	9.5	142.5
TagMe_Vs._Wikimeta	0.011	TagMe	11	14.09	155
		Wikimeta	10	7.6	76
TagMe_Vs._Il.Wikifier	0.297				
TagMe_Vs._Denote	0.796				
Wikipedia Miner_Vs._Wikimeta	0.796				
Wikipedia Miner_Vs._Il.Wikifier	0.005	Wikipedia Miner	15	10.53	158
		Il.wikifier	13	19.08	248
Wikipedia Miner_Vs._Denote	0.687				
Wikimeta_Vs._Il.Wikifier	0.027	Wikimeta	10	8.55	85.5
		Il.wikifier	13	14.65	190.5
Wikimeta_Vs._Denote	0.866				
Il.Wikifier_Vs._Denote	0.033	Il.wikifier	13	15.92	207
		Denote	12	9.83	118

And for the tweets, Table 4.4 shows that TagMe and Illinois Wikifier perform better than the remaining four systems.

Conclusion for Q2 - Case 2: Regarding the system's ability to find the best sense for the annotated phrases correctly, following results has been achieved:

- For long texts, Wikipedia Miner performs significantly better than TagMe, DBPedia Spotlight, Wikimeta and Denote.
- For short texts, TagMe and Wikipedia Miner perform significantly better than the other systems.
- For tweets, TagMe and Illinois Wikifier perform significantly better than the other remaining four systems.

Same two steps are applied for Questions 3-9 since they fall into the same category as ordinal data. Following Tables 4.5 and 4.6 summarize the final result for questions 2-9 for case 1 and case 2 respectively. Not all the systems are listed in Tables 4.5 and 4.6 because they do not show

significant better performance. But Appendix A shows output tables for KW Test and U Test for both cases and for each question.

Case 1 Result – Questions 2-9:

Table 4.5 Summary Table for Questions 2-9 – Case 1

Question	System	Significantly Better Performance	Compared to
2. Ability to correctly find the best sense for the annotated phrases.	Wikipedia Miner	Long Text	Tweets
		Short Text	Tweets
3. The highlighted texts and their related annotations are accurate.	Wikipedia Miner	Long Text	Tweets
		Short Text	Tweets
	DBPedia Spotlight	Long Text	Tweets
		Short Text	Tweets
4. The highlighted texts and their related annotations are relevant.	Wikipedia Miner	Long Text	Tweets
		Short Text	Tweets
	DBPedia Spotlight	Long Text	Tweets
		Short Text	Tweets
5. The annotator was able to identify the main phrases that needed to be annotated.	Wikipedia Miner	Long Text	Tweets
		Short Text	Tweets
	DBPedia Spotlight	Long Text	Tweets
6. The annotations helped me understand the document in ways which would not be otherwise possible.	Wikipedia Miner	Long Text	Tweets
		Short Text	Tweets
7. The annotations were quite informative beyond what was understood from the document alone.	Wikipedia Miner	Long Text	Tweets
		Short Text	
	Denote	Long Text	Tweets
		Short Text	
8. The annotator has been able to identify the central theme of the document correctly.	Wikipedia Miner	Long Text	Tweets
		Short Text	
9. The annotator was able to identify famous people, places and organizations.	Wikipedia Miner	Long Text	Tweets
		Short Text	
	Denote	Long Text	Tweets
		Short Text	

Case 2 Result – Questions 2-9:

Table 4.6 Summary Table for Questions 2-9 – Case 2

Question	For the given type of text	Significantly Better Performance	Compared to
2. Ability to correctly find the best sense for the annotated phrases.	Long Text	Wikipedia Miner	DBPedia Spotlight, TagMe, Wikimeta, Denote
	Short Text	Wikipedia Miner, TagMe	Wikimeta
	Tweet	TagMe, Il. Wikifier	Denote, DBPedia Spotlight, Wikimeta, Wikipedia Miner
3. The highlighted texts and their related annotations are accurate.	Long Text	Wikipedia Miner	DBPedia Spotlight, Il. Wikifier, TagMe, Wikimeta, Denote
	Short Text	Wikipedia Miner	Il.Wikifier, Denote Wikimeta
	Tweet	TagMe, Il.Wikifier	Denote, DBPedia Spotlight, Wikipedia Miner, Wikimeta
4. The highlighted texts and their related annotations are relevant.	Long Text	Wikipedia Miner	Il.Wikifier, DBPedia Spotlight, TagMe, Wikimeta, Denote
	Short Text	TagMe, Wikipedia Miner	Denote, Il.Wikifier, Wikimeta
	Tweet	TagMe, Il.Wikifier	Denote, DBPedia Spotlight, Wikimeta
5. The annotator was able to identify the main phrases that needed to be annotated.	Long Text	Denote, DBPedia Spotlight, TagMe, Wikipedia Miner, Il.Wikifier	Wikimeta
	Short Text	Wikipedia Miner	Wikimeta, Denote, DBPedia Spotlight, Il.Wikifier,
	Tweet	TagMe	Wikimeta, Denote, DBPedia Spotlight, Wikipedia Miner
6. The annotations helped me understand the document in ways which would not be otherwise possible.	Long Text	Wikipedia Miner	Il.Wikifier, DBPedia Spotlight, Wikimeta
	Short Text	TagMe	Denote, II. Wikifier, DBPedia Spotlight, Wikimeta

	Tweet	TagMe, Il.Wikifier	Denote, DBPedia Spotlight, Wikimeta, Wikipedia Miner
7. The annotations were quite informative beyond what was understood from the document alone.	Long Text	Wikipedia Miner	DBPedia Spotlight, Wikimeta, Denote
	Short Text	TagMe	Il.Wikifier, DBPedia Spotlight, Wikimeta
	Tweet	TagMe, Il.Wikifier	Denote, DBPedia Spotlight, Wikimeta, Wikipedia Miner
8. The annotator has been able to identify the central theme of the document correctly.	Long Text	Wikipedia Miner	DBPedia Spotlight, TagMe, Wikimeta, Denote
	Short Text	TagMe	Denote, Il.Wikifier, DBPedia Spotlight, Wikimeta
	Tweet	TagMe	Wikimeta, Denote, DBPedia Spotlight, Wikipedia Miner
9. The annotator was able to identify famous people, places and organizations.	Short Text	TagMe	Denote, Il.Wikifier, DBPedia Spotlight, Wikimeta
	Tweet	TagMe	Denote, Wikipedia Miner, DBPedia Spotlight, Wikimeta

4.2 Analysis on Question 10-12

As described earlier in Chapter 3, Questions 10-12 focus on the comprehensiveness of the system. Questions 10 and 12 have three options while Question 11 has five possible options. Since the difference between the options is not equal, Questions 10, 11 and 12 can also be treated as ordinal data, assigning values to the collected responses as shown in Tables 4.7, 4.8 and 4.9 respectively.

Table 4.7 Values assigned to the options for Question 10

No. of annotations provided	Value assigned
Suitable	3
Too Many	2
Too Few	1

Table 4.8 Values assigned to the options for Question 11

Relevancy	Value assigned
Almost completely relevant	5
Predominantly relevant, with little irrelevancy	4
Roughly half relevant, half irrelevant	3
Predominantly irrelevant, with little relevancy	2
Almost completely irrelevant	1

Table 4.9 Values assigned to the options for Question 12

Response	Value assigned
All the entities I would select and annotate myself	3
Roughly half of the entities I would select and annotate myself	2
None of the entities I would select and annotate myself	1

As Questions 10-12 are considered as ordinal data, they can be analysed the same way that we analysed Questions 2-9 by applying Kruskal Wallis Test to analyse more than two grouping variables and Mann Whitney U Test to compare only two group variables. Appendix A provides the output tables for both cases for Questions 10-12. Tables 4.10 and 4.11 show the summary tables for Questions 10-12 for case 1 and case 2 respectively.

Case 1 Result – Questions 10-12:

Table 4.10 Summary Table for Questions 10-12 – Case 1

Question	System	Significantly Better Performance	Compared to
10.The number of annotations provided for this document by the annotator system was:	Wikipedia Miner	Long Text	Tweets
		Short Text	
	TagMe	Short Text	
11.The annotations produced by the annotation tool are:	Wikipedia Miner	Long Text	Tweets
		Short Text	
12.The annotator recognized and annotated:	Wikipedia Miner	Long Text	Tweets
		Short Text	

As one can see from the Table 4.10, the following result can be concluded for Case 1:

As per the participants' opinion,

- Regarding the number of annotations provided by the system, Wikipedia Miner provides significantly more suitable amount of annotations for long and short texts than tweets, while TagMe gives more suitable amount of annotations for short text compared to tweets.
- Regarding the relevancy of the annotations produced by the system, Wikipedia Miner provides significantly more relevant annotations for long and short texts compared to tweets.
- Selecting and annotating entities that the participant would select and annotate, Wikipedia Miner performs significantly better for long and short texts compared to tweets.

Case 2 Result – Questions 10-12:

Table 4.11 Summary Table for Questions 10-12 – Case 2

Question	For the given type of text	Significantly Better Performance	Compared to
10.The number of annotations provided for this document by the annotator system was:	Tweet	TagMe	Denote, Wikipedia Miner, DBPedia Spotlight
11.The annotations produced by the annotation tool are:	Short Text	TagMe, Wikipedia Miner	Il.Wikifier, Wikimeta
12.The annotator recognized and annotated:	Long Text	Wikipedia Miner	Wikimeta, Denote, DBPedia Spotlight, Il.Wikifier
	Tweet	TagMe	Wikipedia Miner, Il.Wifier

Following result can be concluded for Case 2 for Questions 10-12:

As per the participants' feeling,

- Regarding the number of annotations provided by the system, for the tweets, TagMe performs significantly better and provides more suitable amount of annotations compared to Denote, Wikipedia Miner and DBPedia Spotlight.

- Regarding the relevancy of the annotations produced by the system, for the short text, TagMe and Wikipedia Miner provide significantly more relevant annotations than Il. Wikifier and Wikimeta.
- Selecting and annotating entities that the participant would select and annotate, for long text, Wikipedia Miner performs significantly better than Wikimeta, Denote, DBPedia Spotlight and Il. Wikifier, while for the tweets, TagMe performs significantly better than Wikipedia Miner and Il. Wikifier.

4.3 Analysis on Question 13

Question 13 provides the overall feedback asking how the participants felt analysing the output whether it is manually annotated or machine annotated. Chi square test was considered to be an ideal test to analyse this question [4]. To find out whether the type of text or any system has any impact in deciding the output is manually annotated or machine annotated. For this question also, we have two cases. The analyses for both cases are explained below.

4.3.1 Question 13 - Case 1

Variables: Types of texts and Responses from the participants for a given system

The null hypothesis and alternate hypothesis are as follows:

Null Hypothesis: The proportion of responses, regarding the participants' feeling whether the output is manually annotated or machine annotated, is independent of the type of text for a given system.

Alternative Hypothesis: The proportion of responses, regarding the participants' feeling whether the output is manually annotated or machine annotated, is dependent of the type of text for a given system.

Again, the threshold value decided is 0.05. So if the *p-value* from the chi-square test is less than 0.05, then it shows the significant association between type of text and the responses for a given system.

Just for the illustration purpose, Case 1 – for DBPedia Spotlight has been explained in detail below.

Given System: DBPedia Spotlight

Using SPSS and after applying Chi-Square test, the following result has been produced as shown in Figures 4.17 and 4.18. Figure 4.17 shows the Chi-Square values and Figure 4.18 shows the bar chart comparison between different types of texts for DBPedia Spotlight.

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	2.648[a]	2	.266
Likelihood Ratio	2.890	2	.236
N of Valid Cases	51		

Figure 4.17 Output from SPSS for DBPedia Spotlight – Questions 13 – Case 1

As shown in Figure 4.17, the *p-value* is greater than 0.05, which means, we failed to reject the null hypothesis. There is no significant relation between the response and type of text for DBPedia Spotlight. In other words, if the long text, short text and tweet outputs of DBPedia Spotlight are given to the participants, regardless the type of text; they would not be able to clearly decide whether the outputs are machine annotated or manually annotated.

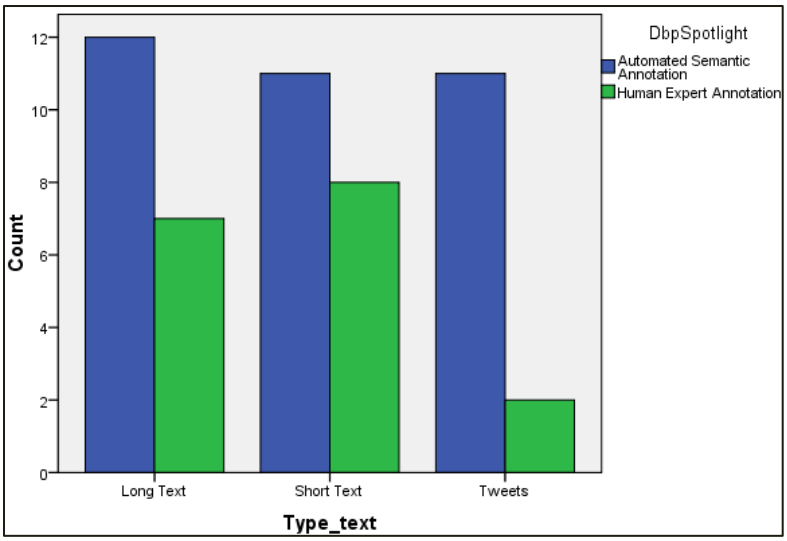

Figure 4.18 Bar chart comparison for DBPedia Spotlight – Question 13 – Case 1

Same treatment can be applied for all the systems. Table 4.12 shows the summary table for Question 13 – Case 1.

Table 4.12 Summary Table for Questions 13 – Case 1

System	Chi-square value	Degrees of freedom	*p-value*
DBPedia Spotlight	2.648	2	.266
TagMe	9.164	2	.010
Wikipedia Miner	5.739	2	.057
Wikimeta	0.626	2	.731
Il.Wikifier	0.754	2	.686
Denote	6.544	2	.038

As seen from Table 4.12, for the systems except TagMe and Denote, the participants are not able to clearly say that the given text is manually annotated or machine annotated. For TagMe and Denote, we reject the null hypothesis. Hence, there is some significant relation between the responses (the participants to decide whether the output is from human annotation or machine annotation) and type of text. So, let's look at the bar charts for TagMe and Denote.

TagMe and Denote bar charts are shown in Figure 4.19 and Figure 4.20 respectively.

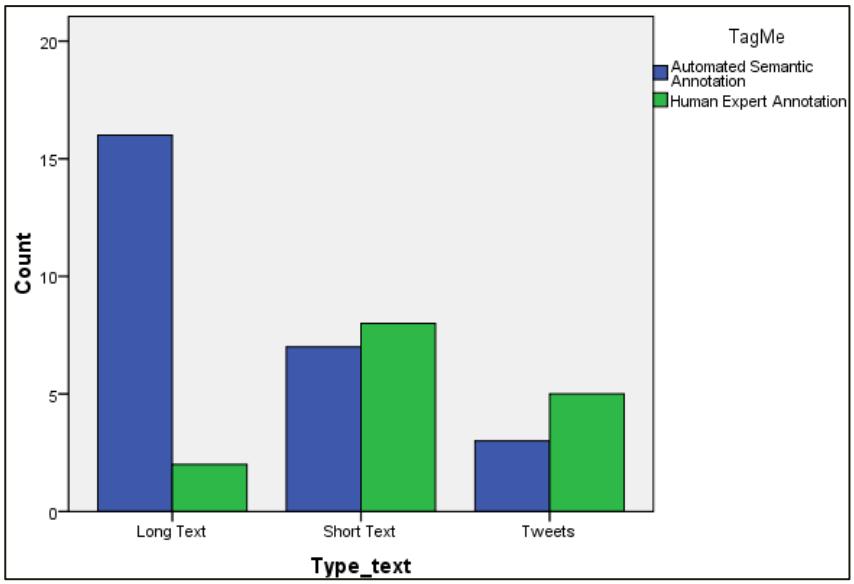

Figure 4.19 Bar chart comparison for TagMe – Question 13 – Case 1

By looking at Figure 4.19, one can notice that the human participants felt that the majority of the long texts are machine annotated. So, if long text, short text and tweet outputs, annotated by TagMe, are provided to the participants, they would be able to identify the long text as machine annotated document. So in terms of performance, TagMe could not give good performance on long text. But for short texts and tweets, the counts for the human annotation category is greater than the machine annotation category which means TagMe performs better on short texts and tweets and provides the output that a human expert may provide.

Same way, Figure 4.20 tells that Denote is not performing well enough on long texts and tweets and the participants are able to clearly state that the given long text or tweet is machine annotated. But for the short text, Denote provides the output that the participant may think the output is manually annotated.

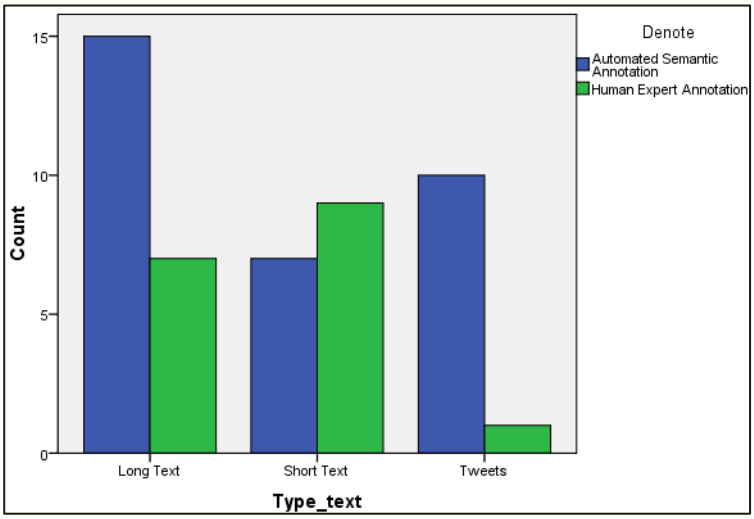

Figure 4.20 Bar chart comparison for Denote – Question 13 – Case 1

4.3.2 Question 13 - Case 2

V*ariables: Systems and Responses from the participants for a given text type*

The null hypothesis and alternate hypothesis are as follows:

> *Null Hypothesis:* The proportion of responses, regarding the participants' feeling whether the output is manually annotated or machine annotated, is independent of the system for a given text type.

Alternative Hypothesis: The proportion of responses, regarding the participants' feeling whether the output is manually annotated or machine annotated, is dependent of the system for a given text type.

By applying the same approach that we have applied in Question 13 – Case 1, the following result, as shown in Table 4.13 has been achieved:

Table 4.13 Summary Table for Questions 13 – Case 2

System	Chi-square value	Degrees of freedom	*p-value*
Long Texts	10.865	5	0.054
Short Texts	4.469	5	0.484
Tweets	11.576	5	0.041

So by looking at Table 4.13, the p-value for the long texts and short texts are greater than 0.05, which means there is no significant relation between the responses (the participants to decide whether the output is from human annotation or machine annotation) and the systems. However, the *p-value* for Tweets is less than 0.05, which means we reject the null hypothesis. There is some significant relation between the response and the systems for given Tweets. Let's look at the bar chart for Tweets.

Figure 4.21 shows the bar chart for tweets – case 2.

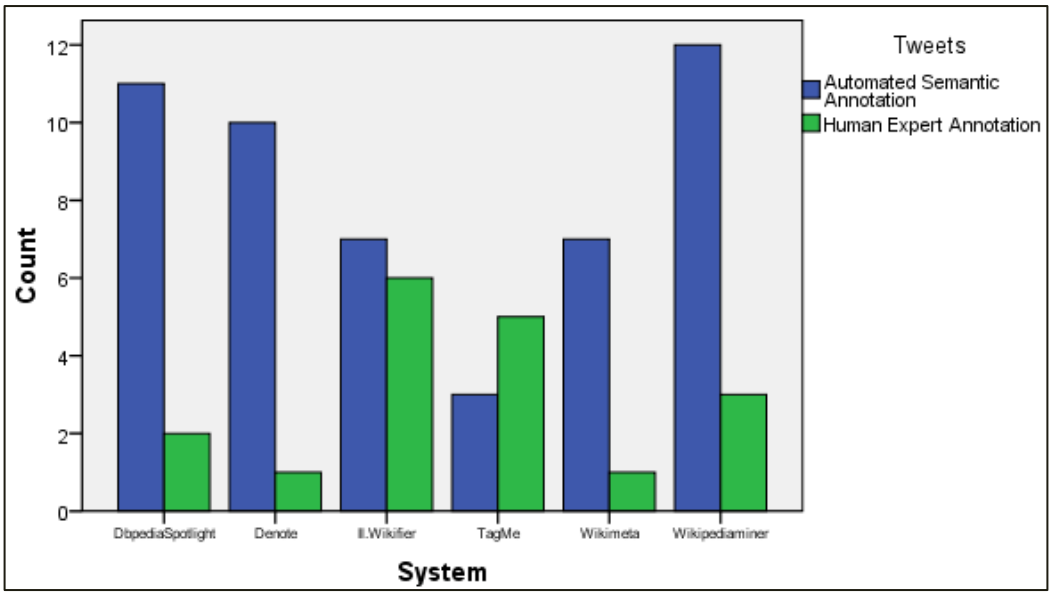

Figure 4.21 Bar chart comparison for Tweets – Question 13 – Case 2

As seen from Figure 4.21, if the same tweet is processed using all these six systems and these outputs are given to the participants, the participants would think that the TagMe output is annotated by human expert while other outputs are machine annotated. This means, TagMe performed pretty well on tweets compared to the other annotation systems and the output is good enough that makes the participants think that the output is manually annotated.

4.4 Chapter Summary

In this chapter, the detailed analysis for each question has been presented. As Likert-type questions can be treated as ordinal data, two statistical tests – KW Test and U Test have been performed to find out which system/systems exhibit significantly different performance. Same way, for the last question, Chi-Square Test has been performed. By checking both cases for each question, the following results can be concluded.

Based on this subjective evaluation, from the participants' feedback:

(a) For the system's ability to find the correct mentions and ability to disambiguate the terms to accurate and relevant articles, Wikipedia Miner and DBPedia Spotlight performed better on long texts and short texts than tweets.

(b) For the system's ability to help the user to better understand the given document by providing more information, correct annotations and links, Wikipedia Miner performed better on long texts and short texts than tweets. Also, for the long and short texts, Denote was able to provide informative annotations beyond what was understood from the document alone.

(c) Wikipedia Miner and Denote performs better in identifying famous people, places and organizations for long texts and short texts than tweets.

(d) For the comprehensiveness of the system, Wikipedia Miner showed better performance on long texts and short texts than tweets. TagMe also provided the suitable amount of annotations for short texts than tweets.

(e) If the user has a long text to annotate, Wikipedia Miner would be a good choice compared to other systems.

(f) If the user has a short text to annotate, TagMe would be a good choice compared to other systems.

(g) If the user has a tweet to annotate, TagMe would be a good choice compared to other systems.

(h) For the named entity recognition for the short texts and tweets, TagMe performs better than other systems.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This research work provides the basic foundation on how the subjective evaluation technique can be applied to evaluate any semantic annotation system. Six different semantic annotation systems, namely Wikipedia Miner, TagMe, Denote, Wikimeta, Illinois Wikifier and DBPedia Spotlight, are evaluated here. 60 participants were involved in the evaluation. The outputs generated from these systems were given to the participants. The responses from the participants about what they think how well the annotators perform was gathered using a survey by asking systematic questions. Their responses were analysed using standard statistical tests. By examining the results, the following conclusion has been determined:

- Wikipedia Miner performs better on long texts compared to other systems.
- TagMe performs better on short texts compared to other systems.
- TagMe performs better on tweets compared to other systems.

The benefit of this technique is that the end users are directly in effect and evaluating the semantic annotation systems. This could help the system designers to better understand what the users want and how could they improve their annotator design. Also, the engagement of various opinions from the users can give more complete and accurate result. Additionally, this approach can be applied and extended to evaluate any semantic annotation system.

5.2 Future Work

As for the future work, this approach can be extended by involving more number of participants. More diversified viewpoints can be collected by involving experts of various fields. More number of participants can provide more accurate result. Also the categories involved can be increased. The analyses could be extended to find the system's performance for each category. Modified questions that can lead to more specific and accurate result would be encouraged. Also,

in this approach, we have assumed that the participants recognized the type of the given document correctly, but more robust approach could be designed to automatically identify and organize the collected responses.

APPENDIX A

Question 3 - Are the highlighted text and their related annotations accurate?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test						
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**Wikimeta**	**Il.Wikifier**	**Denote**
Chi-Square	6.887	4.579	12.175	2.234	4.009	4.48
Df	2	2	2	2	2	2
Asymp. Sig	0.032	0.101	0.002	0.327	0.135	0.106

Case 1 – Step 2 – U Test

Mann-Whitney Test -DBPedia Spotlight					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.851				
Long Text_Vs._Tweets	0.012	Long Text	21	19.93	418.5
		Tweets	12	11.88	142.5
Short Text_Vs._Tweets	0.032	Short Text	20	19.1	382
		Tweets	12	12.17	146

Mann-Whitney Test -Wikipedia Miner					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.648				
Long Text_Vs._Tweets	0.002	Long Text	16	21.34	341.5
		Tweets	16	11.66	186.5
Short Text_Vs._Tweets	0.004	Short Text	20	22.78	455.5
		Tweets	16	13.16	210.5

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	20.106	19.12	20.349
Df	5	5	5
Asymp. Sig	0.001	0.002	0.001

Case 2 – Step 2 – U Test

Mann-Whitney Test -Long Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.672				
Dbp.Spotlight_Vs._Wikipediaminer	0.019	Dbp.Spotlight	21	15.71	330
		Wikipediaminer	16	23.71	373
Dbp.Spotlight_Vs._Wikimeta	0.003	Dbp.Spotlight	21	28.07	589
		Wikimeta	23	17.41	400
Dbp.Spotlight_Vs._Il.wikifier	0.228				
Dbp.Spotlight_Vs._Denote	0.444				
TagMe_Vs._Wikipediaminer	0.04	TagMe	20	15.48	309.5
		Wikipediaminer	23	22.28	356.5

Mann-Whitney Test -Long Text .. *contd.*					
TagMe_Vs._Wikimeta	0.1				
TagMe_Vs._Il.Wikifier	0.198				
TagMe_Vs._Denote	0.848				
Wikipediaminer_Vs._Wikimeta	0	Wikipediaminer	16	28.13	450
		Wikimeta	23	14.35	330
Wikipediaminer_Vs._Il.Wikifier	0.375				
Wikipediaminer_Vs._Denote	0.015	Wikipediaminer	16	23.66	378.5
		Denote	21	15.45	324.5
Wikimeta_Vs._Il.Wikifier	0.002	Wikimeta	23	16.65	383
		Il.Wikifier	20	28.15	563
Wikimeta_Vs._Denote	0.096				
Il.Wikifier_Vs._Denote	0.114				

Mann-Whitney Test -Short Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.259				
Dbp.Spotlight_Vs._Wikipediaminer	0.138				
Dbp.Spotlight_Vs._Wikimeta	0.031	Dbp.Spotlight	20	22.4	448
		Wikimeta	17	15	255
Dbp.Spotlight_Vs._Il.wikifier	0.414				
Dbp.Spotlight_Vs._Denote	0.611				
TagMe_Vs._Wikipediaminer	0.754				
TagMe_Vs._Wikimeta	0.001	TagMe	17	22.74	386.5
		Wikimeta	17	12.26	208.5
TagMe_Vs._Il.Wikifier	0.052				
TagMe_Vs._Denote	0.105				
Wikipediaminer_Vs._Wikimeta	0	Wikipediaminer	20	24.8	496
		Wikimeta	17	12.18	207
Wikipediaminer_Vs._Il.Wikifier	0.016	Wikipediaminer	20	22.7	454
		Il.Wikifier	17	14.65	249
Wikipediaminer_Vs._Denote	0.042	Wikipediaminer	20	21.5	430
		Denote	16	14.75	236
Wikimeta_Vs._Il.Wikifier	0.047	Wikimeta	17	14.29	243
		Il.Wikifier	17	20.71	352
Wikimeta_Vs._Denote	0.085				
Il.Wikifier_Vs._Denote	0.924				

Mann-Whitney Test -Tweets					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.003	Dbp.Spotlight	12	8.08	97
		TagMe	11	16.27	179
Dbp.Spotlight_Vs._Wikipediaminer	0.489				
Dbp.Spotlight_Vs._Wikimeta	0.589				
Dbp.Spotlight_Vs._Il.wikifier	0.005	Dbp.Spotlight	12	8.92	107
		Il.wikifier	13	16.77	218
Dbp.Spotlight_Vs._Denote	0.824				
TagMe_Vs._Wikipediaminer	0.006	TagMe	11	18.82	207
		Wikipediaminer	16	10.69	171
TagMe_Vs._Wikimeta	0.003	TagMe	11	15.45	170
		Wikimeta	11	7.55	83
TagMe_Vs._Il.Wikifier	0.376				
TagMe_Vs._Denote	0.005	TagMe	11	15.27	168
		Denote	11	7.73	85
Wikipediaminer_Vs._Wikimeta	0.293				

Mann-Whitney Test –Tweets …*contd.*					
Wikipediaminer_Vs._Il.Wikifier	0.019	Wikipediaminer	16	11.88	190
		Il.wikifier	13	18.85	245
Wikipediaminer_Vs._Denote	0.345				
Wikimeta_Vs._Il.Wikifier	0.004	Wikimeta	11	8.18	90
		Il.wikifier	13	16.15	210
Wikimeta_Vs._Denote	0.521				
Il.Wikifier_Vs._Denote	0.005	Il.wikifier	13	16.15	210
		Denote	11	8.18	90

4. Are the highlighted texts and their related annotations relevant?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test						
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**Wikimeta**	**Il.Wikifier**	**Denote**
Chi-Square	6.358	1.128	10.933	4.093	4.431	2.14
Df	2	2	2	2	2	2
Asymp. Sig	0.042	0.569	0.004	0.129	0.109	0.343

Case 1 – Step 2 – U Test

Mann-Whitney Test -DBPedia Spotlight					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.765				
Long Text_Vs._Tweets	0.015	Long Text	21	20	420
		Tweets	12	11.75	141
Short Text_Vs._Tweets	0.042	Short Text	20	19.03	380.5
		Tweets	12	12.29	147.5

Mann-Whitney Test -Wikipedia Miner					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.33				
Long Text_Vs._Tweets	0.002	LongText	16	21.75	340
		Tweets	16	11.75	188
Short Text_Vs._Tweets	0.013	Short Text	20	22	440
		Tweets	16	14.13	226

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	19.294	17.67	16.08
df	5	5	5
Asymp. Sig	0.002	0.003	0.007

Case 2 – Step 2 – U Test

Mann-Whitney Test -Long Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.913				
Dbp.Spotlight_Vs._Wikipediaminer	0.027	Dbp.Spotlight	21	15.76	331
		Wikipediaminer	16	23.25	372
Dbp.Spotlight_Vs._Wikimeta	0.062				

Mann-Whitney Test -Long Text …*contd.*					
Dbp.Spotlight_Vs._Il.wikifier	0.294				
Dbp.Spotlight_Vs._Denote	0.411				
TagMe_Vs._Wikipediaminer	0.047	TagMe	20	15.6	312
		Wikipediaminer	16	22.13	354
TagMe_Vs._Wikimeta	0.056				
TagMe_Vs._Il.Wikifier	0.417				
TagMe_Vs._Denote	0.343				
Wikipediaminer_Vs._Wikimeta	0	Wikipediaminer	16	28.28	452.5
		Wikimeta	23	14.24	327.5
Wikipediaminer_Vs._Il.Wikifier	0.172				
Wikipediaminer_Vs._Denote	0.004	Wikipediaminer	16	25.25	404
		Denote	22	15.32	337
Wikimeta_Vs._Il.Wikifier	0.003	Wikimeta	23	16.89	388.5
		Il.Wikifier	20	27.88	557.5
Wikimeta_Vs._Denote	0.336				
Il.Wikifier_Vs._Denote	0.065				

Mann-Whitney Test -Short Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.162				
Dbp.Spotlight_Vs._Wikipediaminer	0.102				
Dbp.Spotlight_Vs._Wikimeta	0.073				
Dbp.Spotlight_Vs._Il.wikifier	0.628				
Dbp.Spotlight_Vs._Denote	0.987				
TagMe_Vs._Wikipediaminer	0.868				
TagMe_Vs._Wikimeta	0.001	TagMe	17	22.74	386.5
		Wikimeta	17	12.26	208.5
TagMe_Vs._Il.Wikifier	0.045	TagMe	17	20.09	341.5
		Il.Wikifier	16	13.72	219.5
TagMe_Vs._Denote	0.143				
Wikipediaminer_Vs._Wikimeta	0	Wikipediaminer	20	24.55	491
		Wikimeta	17	12.47	212
Wikipediaminer_Vs._Il.Wikifier	0.018	Wikipediaminer	20	21.98	439.5
		Il.Wikifier	16	14.16	226.5
Wikipediaminer_Vs._Denote	0.084				
Wikimeta_Vs._Il.Wikifier	0.1				
Wikimeta_Vs._Denote	0.053				
Il.Wikifier_Vs._Denote	0.601				

Mann-Whitney Test -Tweets					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.009	Dbp.Spotlight	12	8.29	99.5
		TagMe	10	15.35	153.5
Dbp.Spotlight_Vs._Wikipediaminer	0.121				
Dbp.Spotlight_Vs._Wikimeta	0.591				
Dbp.Spotlight_Vs._Il.wikifier	0.004	Dbp.Spotlight	12	8.58	103
		Il.Wikifier	12	16.42	197
Dbp.Spotlight_Vs._Denote	0.593				
TagMe_Vs._Wikipediaminer	0.084				
TagMe_Vs._Wikimeta	0.014	TagMe	10	14.4	144
		Wikimeta	11	7.91	87
TagMe_Vs._Il.Wikifier	0.719	TagMe	10	12	120

Mann-Whitney Test –Tweets …*contd.*					
TagMe_Vs._Denote	0.049	TagMe	10	14.4	144
		Denote	12	9.08	109
Wikipediaminer_Vs._Wikimeta	0.131				
Wikipediaminer_Vs._Il.Wikifier	0.119				
Wikipediaminer_Vs._Denote	0.408				
Wikimeta_Vs._Il.Wikifier	0.007	Wikimeta	11	8.09	89
		Il.Wikifier	12	15.58	187
Wikimeta_Vs._Denote	0.283				
Il.Wikifier_Vs._Denote	0.031	Il.Wikifier	12	15.5	186
		Denote	12	9.5	114

5. Was the annotator able to identify the main phrases that needed to be annotated?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test					
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**WikiMeta**	**II.Wikifier**
Chi-Square	7.222	0.935	12.149	0.307	4.159
df	2	2	2	2	2
Asymp. Sig	0.027	0.627	0.002	0.858	0.125

Case 1 – Step 2 – U Test

Mann-Whitney Test -DBPedia Spotlight					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.136				
Long Text_Vs._Tweets	0.011	Long Text	20	20.33	406.5
		Tweets	13	11.88	154.5
Short Text_Vs._Tweets	0.139				

Mann-Whitney Test -DBPedia Spotlight					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.596				
Long Text_Vs._Tweets	0.005	LongText	16	21	336
		Tweets	16	12	192
Short Text_Vs._Tweets	0.002	Short Text	20	23.25	465
		Tweets	16	12.56	201

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	13.697	16.292	16.56
df	5	5	5
Asymp. Sig	0.018	0.006	0.005

Case 2 – Step 2 – U Test

Mann-Whitney Test -Long Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.321				
Dbp.Spotlight_Vs._Wikipediaminer	0.229				
Dbp.Spotlight_Vs._Wikimeta	0.058				

Mann-Whitney Test -Long Text …*contd.*					
Dbp.Spotlight_Vs._Il.wikifier	0.275				
Dbp.Spotlight_Vs._Denote	0.419				
TagMe_Vs._Wikipediaminer	0.973				
TagMe_Vs._Wikimeta	0.009	TagMe	20	27.15	543
		Wikimeta	23	17.52	403
TagMe_Vs._Il.Wikifier	0.954				
TagMe_Vs._Denote	0.083				
Wikipediaminer_Vs._Wikimeta	0.005	Wikipediaminer	16	25.78	412.5
		Wikimeta	23	15.98	367.5
Wikipediaminer_Vs._Il.Wikifier	0.932				
Wikipediaminer_Vs._Denote	0.07				
Wikimeta_Vs._Il.Wikifier	0.008	Wikimeta	23	17.46	401.5
		Il.Wikifier	20	27.23	544.5
Wikimeta_Vs._Denote	0.384				
Il.Wikifier_Vs._Denote	0.068				

Mann-Whitney Test -Short Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.052				
Dbp.Spotlight_Vs._Wikipediaminer	0.003	Dbp.Spotlight	20	15.3	306
		Wikipediaminer	20	25.7	514
Dbp.Spotlight_Vs._Wikimeta	0.364				
Dbp.Spotlight_Vs._Il.wikifier	0.767				
Dbp.Spotlight_Vs._Denote	0.766				
TagMe_Vs._Wikipediaminer	0.685				
TagMe_Vs._Wikimeta	0.035	TagMe	17	21	357
		Wikimeta	17	14	238
TagMe_Vs._Il.Wikifier	0.12				
TagMe_Vs._Denote	0.085				
Wikipediaminer_Vs._Wikimeta	0.004	Wikipediaminer	20	23.6	472
		Wikimeta	17	13.59	231
Wikipediaminer_Vs._Il.Wikifier	0.012	Wikipediaminer	20	22.95	459
		Il.Wikifier	17	14.35	244
Wikipediaminer_Vs._Denote	0.008	Wikipediaminer	20	22.43	448.5
		Denote	16	13.59	217.5
Wikimeta_Vs._Il.Wikifier	0.216				
Wikimeta_Vs._Denote	0.309				
Il.Wikifier_Vs._Denote	0.984				

Mann-Whitney Test -Tweets					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.001	Dbp.Spotlight	13	8.31	108
		TagMe	11	17.45	192
Dbp.Spotlight_Vs._Wikipediaminer	0.91				
Dbp.Spotlight_Vs._Wikimeta	0.656				
Dbp.Spotlight_Vs._Il.wikifier	0.012	Dbp.Spotlight	13	9.85	128
		Il.wikifier	13	17.15	223
Dbp.Spotlight_Vs._Denote	0.845				
TagMe_Vs._Wikipediaminer	0.003	TagMe	11	19.41	213.5
		Wikipediaminer	16	10.28	164.5
TagMe_Vs._Wikimeta	0.034	TagMe	11	14.32	157.5
		Wikimeta	11	8.68	95.5

Mann-Whitney Test –Tweets …*contd.*					
TagMe_Vs._Il.Wikifier	0.248				
TagMe_Vs._Denote	0.011	TagMe	11	15.64	172
		Denote	12	8.67	104
Wikipediaminer_Vs._Wikimeta	0.668				
Wikipediaminer_Vs._Il.Wikifier	0.026	Wikipediaminer	16	11.91	190.5
		Il.wikifier	13	18.81	244.5
Wikipediaminer_Vs._Denote	0.905				
Wikimeta_Vs._Il.Wikifier	0.103				
Wikimeta_Vs._Denote	0.801				
Il.Wikifier_Vs._Denote	0.055				

6. Did the annotations help you to understand the document in ways which would not be otherwise possible?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test						
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**WikiMeta**	**II.Wikifier**	**Denote**
Chi-Square	3.521	3.956	8.355	3.62	0.71	5.98
Df	2	2	2	2	2	2
Asymp. Sig	0.172	0.138	0.015	0.164	0.701	0.05

Case 1 – Step 2 – U Test

Mann-Whitney Test -Wikipedia Miner					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.346				
Long Text_Vs._Tweets	0.007	Long Text	16	20.81	333
		Tweets	16	12.19	195
Short Text_Vs._Tweets	0.033	Short Text	19	21.32	405
		Tweets	16	14.06	225

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	14.581	11.558	15.572
df	5	5	5
Asymp. Sig	0.012	0.041	0.008

Case 2 – Step 2 – U Test

Mann-Whitney Test -Long Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.437				
Dbp.Spotlight_Vs._Wikipediaminer	0.019	Dbp.Spotlight	21	15.52	326
		Wikipediaminer	16	23.56	377
Dbp.Spotlight_Vs._Wikimeta	0.055				
Dbp.Spotlight_Vs._Il.wikifier	0.805				
Dbp.Spotlight_Vs._Denote	0.509				
TagMe_Vs._Wikipediaminer	0.146				

Mann-Whitney Test -Long Text …*contd.*					
TagMe_Vs._Wikimeta	0.027	TagMe	20	25.13	502.5
		Wikimeta	21	17.07	358.5
TagMe_Vs._Il.Wikifier	0.612				
TagMe_Vs._Denote	0.906				
Wikipediaminer_Vs._Wikimeta	0.001	Wikipediaminer	16	25.47	407.5
		Wikimeta	21	14.07	295.5
Wikipediaminer_Vs._Il.Wikifier	0.044	Wikipediaminer	16	22.28	356.5
		Il.Wikifier	20	15.48	309.5
Wikipediaminer_Vs._Denote	0.107				
Wikimeta_Vs._Il.Wikifier	0.042	Wikimeta	21	17.4	365.5
Wikimeta_Vs._Il.Wikifier	0.042	Il.Wikifier	20	24.78	495.5
Wikimeta_Vs._Denote	0.024	Wikimeta	21	17.69	371.5
		Denote	22	26.11	574.5
Il.Wikifier_Vs._Denote	0.694				

Mann-Whitney Test -Short Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.004	Dbp.Spotlight	20	14.48	289.5
		TagMe	17	24.32	413.5
Dbp.Spotlight_Vs._Wikipediaminer	0.055				
Dbp.Spotlight_Vs._Wikimeta	0.356				
Dbp.Spotlight_Vs._Il.wikifier	0.275				
Dbp.Spotlight_Vs._Denote	0.227				
TagMe_Vs._Wikipediaminer	0.332				
TagMe_Vs._Wikimeta	0.021	TagMe	17	21.26	361.5
		Wikimeta	17	13.74	233.5
TagMe_Vs._Il.Wikifier	0.028	TagMe	17	21.09	358.5
		Il.wikifier	17	13.91	236.5
TagMe_Vs._Denote	0.044	TagMe	17	20.09	341.5
		Denote	16	13.72	219.5
Wikipediaminer_Vs._Wikimeta	0.236				
Wikipediaminer_Vs._Il.Wikifier	0.282				
Wikipediaminer_Vs._Denote	0.405				
Wikimeta_Vs._Il.Wikifier	0.971				
Wikimeta_Vs._Denote	0.688				
Il.Wikifier_Vs._Denote	0.642				

Mann-Whitney Test -Tweets					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.005	Dbp.Spotlight	13	8.88	115.5
		TagMe	11	16.77	184.5
Dbp.Spotlight_Vs._Wikipediaminer	0.787				
Dbp.Spotlight_Vs._Wikimeta	0.812				
Dbp.Spotlight_Vs._Il.wikifier	0.039	Dbp.Spotlight	13	10.5	136.5
		Il.wikifier	13	16.5	214.5
Dbp.Spotlight_Vs._Denote	0.909				
TagMe_Vs._Wikipediaminer	0.007	TagMe	11	18.82	207
		Wikipediaminer	16	10.69	171
TagMe_Vs._Wikimeta	0.012	TagMe	11	14.82	163
		Wikimeta	11	8.18	90
TagMe_Vs._Il.Wikifier	0.262				

Mann-Whitney Test –Tweets …*contd.*					
TagMe_Vs._Denote	0.006	TagMe	11	15.95	175
		Denote	12	8.38	100.5
Wikipediaminer_Vs._Wikimeta	0.723				
Wikipediaminer_Vs._Il.Wikifier	0.054				
Wikipediaminer_Vs._Denote	0.775				
Wikimeta_Vs._Il.Wikifier	0.035	Wikimeta	11	9.27	102
		Il.wikifier	13	15.23	198
Wikimeta_Vs._Denote	0.776				
Il.Wikifier_Vs._Denote	0.03	Il.wikifier	13	15.96	207.5
		Denote	12	9.79	117.5

7. Were the annotations informative beyond what was understood from the document alone?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test						
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**WikiMeta**	**II.Wikifier**	**Denote**
Chi-Square	3.733	5.114	14.757	2.334	4.072	6.942
Df	2	2	2	2	2	2
Asymp. Sig	0.155	0.078	0.001	0.311	0.131	0.031

Case 1 – Step 2 – U Test

Mann-Whitney Test -Wikipedia Miner					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.212				
Long Text_Vs._Tweets	0.001	Long Text	15	21.63	324.5
		Tweets	16	10.72	171.5
Short Text_Vs._Tweets	0.002	Short Text	20	23.1	462
		Tweets	16	12.75	204

Mann-Whitney Test -Denote					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.382				
Long Text_Vs._Tweets	0.058				
Short Text_Vs._Tweets	0.01	Short Text	16	17.81	285
		Tweets	12	10.08	121

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	14.581	11.558	15.572
df	5	5	5
Asymp. Sig	0.012	0.041	0.008

Case 2 – Step 2 – U Test

Mann-Whitney Test -Long Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.378				
Dbp.Spotlight_Vs._Wikipediaminer	0.017	Dbp.Spotlight	21	15.1	317
		Wikipediaminer	15	23.27	349
Dbp.Spotlight_Vs._Wikimeta	0.055	Dbp.Spotlight	21	26.24	551
		Wikimeta	23	19.09	439
Dbp.Spotlight_Vs._Il.wikifier	0.191				
Dbp.Spotlight_Vs._Denote	0.881				
TagMe_Vs._Wikipediaminer	0.087	TagMe			
TagMe_Vs._Wikimeta	0.009	TagMe	20	27.15	543
		Wikimeta	23	17.52	403
TagMe_Vs._Il.Wikifier	0.687				
TagMe_Vs._Denote	0.352				
Wikipediaminer_Vs._Wikimeta	0.001	Wikipediaminer	15	26.8	402
		Wikimeta	23	14.74	339
Wikipediaminer_Vs._Il.Wikifier	0.12	Wikipediaminer	15	20.93	314
		Il.Wikifier	20	15.8	316
Wikipediaminer_Vs._Denote	0.023	Wikipediaminer	15	22.37	335.5
		Denote	20	14.73	294
Wikimeta_Vs._Il.Wikifier	0.004	Wikimeta	23	17	391
Wikimeta_Vs._Denote	0.113				
Il.Wikifier_Vs._Denote	0.2				

Mann-Whitney Test -Short Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.049	Dbp.Spotlight	20	15.9	318
		TagMe	17	22.65	385
Dbp.Spotlight_Vs._Wikipediaminer	0.243				
Dbp.Spotlight_Vs._Wikimeta	0.297				
Dbp.Spotlight_Vs._Il.wikifier	0.626				
Dbp.Spotlight_Vs._Denote	0.489				
TagMe_Vs._Wikipediaminer	0.352				
TagMe_Vs._Wikimeta	0.004	TagMe	17	22.26	378.5
		Wikimeta	17	12.74	216.5
TagMe_Vs._Il.Wikifier	0.006	TagMe	17	21.91	372.5
		Il.Wikifier	17	13.09	222.5
TagMe_Vs._Denote	0.117				
Wikipediaminer_Vs._Wikimeta	0.021	Wikipediaminer	20	22.65	453
		Wikimeta	17	14.71	250
Wikipediaminer_Vs._Il.Wikifier	0.071				
Wikipediaminer_Vs._Denote	0.67				
Wikimeta_Vs._Il.Wikifier	0.265				
Wikimeta_Vs._Denote	0.054				
Il.Wikifier_Vs._Denote	0.064				

Mann-Whitney Test -Tweets					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.002	Dbp.Spotlight	13	8.58	111.5
		TagMe	11	17.14	188.5
Dbp.Spotlight_Vs._Wikipediaminer	0.766				
Dbp.Spotlight_Vs._Wikimeta	0.385				
Dbp.Spotlight_Vs._Il.wikifier	0.023	Dbp.Spotlight	13	10.19	132.5
		Il.wikifier	13	16.81	218.5

Mann-Whitney Test –Tweets …*contd.*					
Dbp.Spotlight_Vs._Denote	0.551				
TagMe_Vs._Wikipediaminer	0.001	TagMe	11	19.91	219
		Wikipediaminer	16	9.94	159
TagMe_Vs._Wikimeta	0.003	TagMe	11	15.41	169.5
		Wikimeta	11	7.59	83.5
TagMe_Vs._Il.Wikifier	0.182				
TagMe_Vs._Denote	0.004	TagMe	11	16.14	177.5
		Denote	12	8.21	98.5
Wikipediaminer_Vs._Wikimeta	0.576				
Wikipediaminer_Vs._Il.Wikifier	0.01	Wikipediaminer	16	11.41	182.5
		Il.wikifier	13	19.42	252.5
Wikipediaminer_Vs._Denote	0.792				
Wikimeta_Vs._Il.Wikifier	0.01	Wikimeta	11	8.55	94
		Il.wikifier	13	15.85	206
Wikimeta_Vs._Denote	0.727				
Il.Wikifier_Vs._Denote	0.014	Il.wikifier	13	16.38	213
		Denote	12	9.33	112

8. Has the annotator been able to identify the central theme of the document correctly?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test						
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**WikiMeta**	**II.Wikifier**	**Denote**
Chi-Square	4.412	4.35	13.064	1.92	0.221	2.515
df	2	2	2	2	2	2
Asymp. Sig	0.11	0.114	0.001	0.383	0.896	0.284

Case 1 – Step 2 – U Test

Mann-Whitney Test -Wikipedia Miner					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.48				
Long Text_Vs._Tweets	0.001	Long Text	16	21.66	346.5
		Tweets	16	11.34	181.5
Short Text_Vs._Tweets	0.004	Short Text	20	22.9	458
		Tweets	16	13	208

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	11.923	16.967	15.27
df	5	5	5
Asymp. Sig	0.036	0.005	0.009

Case 2 – Step 2 – U Test

Mann-Whitney Test -Long Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.52				
Dbp.Spotlight_Vs._Wikipediaminer	0.017	Dbp.Spotlight	20	14.93	298
		Wikipediaminer	16	22.97	367.5

Mann-Whitney Test -Long Text …*contd.*					
Dbp.Spotlight_Vs._Wikimeta	0.035	Dbp.Spotlight	20	25.53	510.5
		Wikimeta	22	17.84	392.5
Dbp.Spotlight_Vs._Il.wikifier	0.273				
Dbp.Spotlight_Vs._Denote	0.955				
TagMe_Vs._Wikipediaminer	0.105	TagMe	19	15.61	296.5
TagMe_Vs._Wikimeta	0.011	TagMe	19	25.89	492
		Wikimeta	22	16.77	369
TagMe_Vs._Il.Wikifier	0.778				
TagMe_Vs._Denote	0.5				
Wikipediaminer_Vs._Wikimeta	0	Wikipediaminer	16	27.69	443
		Wikimeta	22	13.55	298
Wikipediaminer_Vs._Il.Wikifier	0.106	Wikipediaminer	16	21.44	343
		Il.Wikifier	20	16.15	323
Wikipediaminer_Vs._Denote	0.021	Wikipediaminer	16	22.78	364.5
		Denote	20	15.08	301.5
Wikimeta_Vs._Il.Wikifier	0.001	Wikimeta	22	15.98	351.5
		Il.Wikifier	20	27.58	551.5
Wikimeta_Vs._Denote	0.056	Wikimeta	22	18.18	400
		Denote	20	25.15	503
Il.Wikifier_Vs._Denote	0.287				

Mann-Whitney Test -Short Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.01	Dbp.Spotlight	20	15	300
		TagMe	17	23.71	403
Dbp.Spotlight_Vs._Wikipediaminer	0.063				
Dbp.Spotlight_Vs._Wikimeta	0.487				
Dbp.Spotlight_Vs._Il.wikifier	0.373				
Dbp.Spotlight_Vs._Denote	0.893				
TagMe_Vs._Wikipediaminer	0.477				
TagMe_Vs._Wikimeta	0.001	TagMe	17	22.82	388
		Wikimeta	17	12.18	207
TagMe_Vs._Il.Wikifier	0.032	TagMe	17	20.85	354.5
		Il.Wikifier	17	14.15	240.5
TagMe_Vs._Denote	0.001	TagMe	17	21.65	368
		Denote	16	12.06	193
Wikipediaminer_Vs._Wikimeta	0.01	Wikipediaminer	20	23.05	461
		Wikimeta	17	14.24	242
Wikipediaminer_Vs._Il.Wikifier	0.244				
Wikipediaminer_Vs._Denote	0.035	Wikipediaminer	20	21.6	432
		Denote	16	14.63	234
Wikimeta_Vs._Il.Wikifier	0.086				
Wikimeta_Vs._Denote	0.347				
Il.Wikifier_Vs._Denote	0.337				

Mann-Whitney Test -Tweets					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.003	Dbp.Spotlight	13	8.65	112.5
		TagMe	11	17.05	187.5
Dbp.Spotlight_Vs._Wikipediaminer	0.803				
Dbp.Spotlight_Vs._Wikimeta	0.834				
Dbp.Spotlight_Vs._Il.wikifier	0.046	Dbp.Spotlight	13	10.58	137.5
		Il.wikifier	13	16.42	213.5
Dbp.Spotlight_Vs._Denote	0.78				

Mann-Whitney Test –Tweets …*contd.*					
TagMe_Vs._Wikipediaminer	0.003	TagMe	11	19.36	213
		Wikipediaminer	16	10.31	165
TagMe_Vs._Wikimeta	0.007	TagMe	11	15.05	165.5
		Wikimeta	11	7.95	87.5
TagMe_Vs._Il.Wikifier	0.17				
TagMe_Vs._Denote	0.005	TagMe	11	16.05	176.5
		Denote	12	8.29	99.5
Wikipediaminer_Vs._Wikimeta	0.741				
Wikipediaminer_Vs._Il.Wikifier	0.065				
Wikipediaminer_Vs._Denote	0.924				
Wikimeta_Vs._Il.Wikifier	0.049	Wikimeta	11	9.5	104.5
		Il.wikifier	13	15.04	195.5
Wikimeta_Vs._Denote	0.614				
Il.Wikifier_Vs._Denote	0.099				

9. Was the annotator able to identify famous people, places and organizations?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test						
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**WikiMeta**	**II.Wikifier**	**Denote**
Chi-Square	5.327	2.561	6.972	2.033	2.106	7.214
df	2	2	2	2	2	2
Asymp. Sig	0.07	0.278	0.031	0.362	0.349	0.027

Case 1 – Step 2 – U Test

Mann-Whitney Test -Wikipedia Miner					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.958				
Long Text_Vs._Tweets	0.025	Long Text	16	20.13	322
		Tweets	16	12.88	206
Short Text_Vs._Tweets	0.021	Short Text	19	21.55	409.5
		Tweets	16	13.78	220.5

Mann-Whitney Test -Denote					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.319				
Long Text_Vs._Tweets	0.014	LongText	21	20	420
		Tweets	12	11.75	141
Short Text_Vs._Tweets	0.052				

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	7.122	15.031	15.147
df	5	5	5
Asymp. Sig	0.212	0.01	0.01

Case 2 – Step 2 – U Test

Mann-Whitney Test -Short Text					
Systems pairs	**Asymp. Sig**	**System**		**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.02	Dbp.Spotlight	19	14.47	275
		TagMe	16	22.19	355
Dbp.Spotlight_Vs._Wikipediaminer	0.058				
Dbp.Spotlight_Vs._Wikimeta	0.577				
Dbp.Spotlight_Vs._Il.wikifier	0.6				
Dbp.Spotlight_Vs._Denote	0.375				
TagMe_Vs._Wikipediaminer	0.489				
TagMe_Vs._Wikimeta	0.003	TagMe	16	21.97	351.5
		Wikimeta	17	12.32	209.5
TagMe_Vs._Il.Wikifier	0.036	TagMe	16	20.47	327.5
		Il.Wikifier	17	13.74	233.5
TagMe_Vs._Denote	0.016	TagMe	16	19.38	310
		Denote	15	12.4	186
Wikipediaminer_Vs._Wikimeta	0.01	Wikipediaminer	19	22.16	429.5
		Wikimeta	17	13.91	236.5
Wikipediaminer_Vs._Il.Wikifier	0.124				
Wikipediaminer_Vs._Denote	0.139				
Wikimeta_Vs._Il.Wikifier	0.253				
Wikimeta_Vs._Denote	0.072				
Il.Wikifier_Vs._Denote	0.62				

Mann-Whitney Test -Tweets					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.001	Dbp.Spotlight	13	8.38	109
		TagMe	11	17.36	191
Dbp.Spotlight_Vs._Wikipediaminer	0.573				
Dbp.Spotlight_Vs._Wikimeta	0.976				
Dbp.Spotlight_Vs._Il.wikifier	0.028	Dbp.Spotlight	13	10.31	134
		Il.wikifier	13	16.39	217
Dbp.Spotlight_Vs._Denote	0.755				
TagMe_Vs._Wikipediaminer	0.008	TagMe	11	18.77	206.5
		Wikipediaminer	16	10.72	171.5
TagMe_Vs._Wikimeta	0.006	TagMe	11	15.09	166
		Wikimeta	11	7.91	87
TagMe_Vs._Il.Wikifier	0.156				
TagMe_Vs._Denote	0.009	TagMe	11	15.64	172
		Denote	12	8.67	104
Wikipediaminer_Vs._Wikimeta	0.782				
Wikipediaminer_Vs._Il.Wikifier	0.132				
Wikipediaminer_Vs._Denote	0.887				
Wikimeta_Vs._Il.Wikifier	0.075				
Wikimeta_Vs._Denote	0.773				
Il.Wikifier_Vs._Denote	0.11				

10. Was the number of annotations provided for this document by the annotator system in enough amounts?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test						
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**WikiMeta**	**II.Wikifier**	**Denote**
Chi-Square	4.892	9.628	11.921	3.688	2.259	0.649
df	2	2	2	2	2	2
Asymp. Sig	0.087	0.008	0.003	0.158	0.323	0.723

Case 1 – Step 2 – U Test

Mann-Whitney Test -TagMe					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.002	Long Text	20	15.18	303.5
		Short Text	17	23.5	399.5
Long Text_Vs._Tweets	0.74				
Short Text_Vs._Tweets	0.009	Short Text	17	16.5	280.5
		Tweets	11	11.41	125.5

Mann-Whitney Test -Wikipedia Miner					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.656				
Long Text_Vs._Tweets	0.002	LongText	16	21.03	336.5
		Tweets	16	11.97	191.5
Short Text_Vs._Tweets	0.005	Short Text	20	22.43	448.5
		Tweets	16	13.59	217.5

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	8.173	8.54	11.324
df	5	5	5
Asymp. Sig	0.147	0.129	0.045

Case 2 – Step 2 – U Test

Mann-Whitney Test -Tweets					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.031	Dbp.Spotlight	12	9.33	112
		TagMe	11	14.91	164
Dbp.Spotlight_Vs._Wikipediaminer	0.909				
Dbp.Spotlight_Vs._Wikimeta	0.079				
Dbp.Spotlight_Vs._Il.wikifier	0.109				
Dbp.Spotlight_Vs._Denote	0.065				
TagMe_Vs._Wikipediaminer	0.014	TagMe	11	18.09	199
		Wikipediaminer	16	11.19	179
TagMe_Vs._Wikimeta	0.745				
TagMe_Vs._Il.Wikifier	0.638				
TagMe_Vs._Denote	0.371				
Wikipediaminer_Vs._Wikimeta	0.045	Wikipediaminer	16	11.38	182
		Wikimeta	10	16.9	169
Wikipediaminer_Vs._Il.Wikifier	0.065				

Mann-Whitney Test –Tweets …*contd.*					
Wikipediaminer_Vs._Denote	0.032	Wikipediaminer	16	11.81	189
		Denote	12	1808	217
Wikimeta_Vs._Il.Wikifier	0.887				
Wikimeta_Vs._Denote	0.663				
Il.Wikifier_Vs._Denote	0.811				

11. Were the annotations produced by the annotation tool relevant to the topic?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test						
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**WikiMeta**	**II.Wikifier**	**Denote**
Chi-Square	3.592	3.058	10.078	2.176	0.631	1.236
df	2	2	2	2	2	2
Asymp. Sig	0.139	0.217	0.006	0.337	0.729	0.539

Case 1 – Step 2 – U Test

Mann-Whitney Test -Wikipedia Miner					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.321				
Long Text_Vs._Tweets	0.006	Long Text	16	19.38	310
		Tweets	14	11.07	155
Short Text_Vs._Tweets	0.008	Short Text	20	21.13	422.5
		Tweets	14	12.32	172.5

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	13.425	15.362	9.068
Df	5	5	5
Asymp. Sig	0.02	0.009	0.106

Case 2 – Step 2 – U Test

Mann-Whitney Test -Long Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.883				
Dbp.Spotlight_Vs._Wikipediaminer	0.049	Dbp.Spotlight	21	16.14	339
		Wikipediaminer	16	22.75	364
Dbp.Spotlight_Vs._Wikimeta	0.07				
Dbp.Spotlight_Vs._Il.wikifier	0.766				
Dbp.Spotlight_Vs._Denote	0.814				
TagMe_Vs._Wikipediaminer	0.026				
TagMe_Vs._Wikimeta	0.053	TagMe	20	25.18	517.5
		Wikimeta	23	18.63	428.5
TagMe_Vs._Il.Wikifier	0.615				
TagMe_Vs._Denote	0.746				
Wikipediaminer_Vs._Wikimeta	0.001	Wikipediaminer	16	27	432
		Wikimeta	23	15.13	348
Wikipediaminer_Vs._Il.Wikifier	0.075				
Wikipediaminer_Vs._Denote	0.081				

Mann-Whitney Test -Long Text …*contd.*					
Wikimeta_Vs._Il.Wikifier	0.034	Wikimeta	23	18.3	421
		Il.Wikifier	20	26.25	525
Wikimeta_Vs._Denote	0.07				
Il.Wikifier_Vs._Denote	0.989				

Mann-Whitney Test -Short Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.133				
Dbp.Spotlight_Vs._Wikipediaminer	0.139				
Dbp.Spotlight_Vs._Wikimeta	0.088				
Dbp.Spotlight_Vs._Il.wikifier	0.601				
Dbp.Spotlight_Vs._Denote	0.487				
TagMe_Vs._Wikipediaminer	0.79				
TagMe_Vs._Wikimeta	0.002	TagMe	17	22.74	386.5
		Wikimeta	17	12.26	208.5
TagMe_Vs._Il.Wikifier	0.031	TagMe	17	21.03	357.5
		Il.Wikifier	17	13.97	237.5
TagMe_Vs._Denote	0.664				
Wikipediaminer_Vs._Wikimeta	0.001	Wikipediaminer	20	24.45	489
		Wikimeta	17	12.59	214
Wikipediaminer_Vs._Il.Wikifier	0.025	Wikipediaminer	20	22.53	450.5
		Il.Wikifier	17	18.45	252.5
Wikipediaminer_Vs._Denote	0.731				
Wikimeta_Vs._Il.Wikifier	0.173				
Wikimeta_Vs._Denote	0.071				
Il.Wikifier_Vs._Denote	0.33				

12. Would you select and annotate the same annotations that system has provided?

Case 1 – Step 1 – KW Test

Kruskal-Wallis Test						
	Dbp.Spotlight	**TagMe**	**Wikipediaminer**	**WikiMeta**	**II.Wikifier**	**Denote**
Chi-Square	2.08	0.299	12.357	1.048	1.567	3.671
df	2	2	2	2	2	2
Asymp. Sig	0.353	0.861	0.002	0.592	0.457	0.16

Case 1 – Step 2 – U Test

Mann-Whitney Test -Wikipedia Miner					
Text_Type pairs	**Asymp. Sig**	**Text_Type**	**N**	**Mean Rank**	**Sum Of Ranks**
Long Text_Vs._Short Text	0.089				
Long Text_Vs._Tweets	0.002	Long Text	16	21.34	341.5
		Tweets	16	11.66	186.5
Short Text_Vs._Tweets	0.013	Short Text	19	21.61	410.5
		Tweets	16	13.72	219.5

Case 2 – Step 1 – KW Test

Kruskal-Wallis Test			
	Long Text	**Short Text**	**Tweets**
Chi-Square	13.313	5.005	12.981
Df	5	5	5
Asymp. Sig	0.021	0.415	0.024

Case 2 – Step 2 – U Test

Mann-Whitney Test -Long Text					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.03	Dbp.Spotlight	21	17.07	358.5
		TagMe	19	24.29	461.5
Dbp.Spotlight_Vs._Wikipediaminer	0.004	Dbp.Spotlight	21	15	315
		Wikipediaminer	16	24.25	388
Dbp.Spotlight_Vs._Wikimeta	0.488				
Dbp.Spotlight_Vs._Il.wikifier	0.546				
Dbp.Spotlight_Vs._Denote	0.454				
TagMe_Vs._Wikipediaminer	0.34				
TagMe_Vs._Wikimeta	0.048	TagMe	19	24.42	464
		Wikimeta	22	18.05	397
TagMe_Vs._Il.Wikifier	0.156				
TagMe_Vs._Denote	0.13				
Wikipediaminer_Vs._Wikimeta	0.005	Wikipediaminer	16	24.72	395.5
		Wikimeta	22	15.7	345.5
Wikipediaminer_Vs._Il.Wikifier	0.032	Wikipediaminer	16	21.63	346
		Il.Wikifier	19	14.95	284
Wikipediaminer_Vs._Denote	0.021	Wikipediaminer	16	23.19	371
		Denote	21	15.81	332
Wikimeta_Vs._Il.Wikifier	0.902				
Wikimeta_Vs._Denote	0.824				
Il.Wikifier_Vs._Denote	0.952				

Mann-Whitney Test –Tweets					
Systems pairs	**Asymp. Sig**	**System**	**N**	**Mean Rank**	**Sum Of Ranks**
Dbp.Spotlight_Vs._Tagme	0.011	Dbp.Spotlight	13	9.42	122.5
		TagMe	11	16.14	177.5
Dbp.Spotlight_Vs._Wikipediaminer	0.475				
Dbp.Spotlight_Vs._Wikimeta	0.536				
Dbp.Spotlight_Vs._Il.wikifier	0.076				
Dbp.Spotlight_Vs._Denote	0.362				
TagMe_Vs._Wikipediaminer	0.003	TagMe	11	19.14	210.5
		Wikipediaminer	16	10.47	167.5
TagMe_Vs._Wikimeta	0.117				
TagMe_Vs._Il.Wikifier	0.417				
TagMe_Vs._Denote	0.055	TagMe	11	14.5	159.5
		Denote	12	9.71	116.5
Wikipediaminer_Vs._Wikimeta	0.249				
Wikipediaminer_Vs._Il.Wikifier	0.021	Wikipediaminer	16	11.59	185.5
		Il.wikifier	12	18.38	220.5
Wikipediaminer_Vs._Denote	0.112				
Wikimeta_Vs._Il.Wikifier	0.379				
Wikimeta_Vs._Denote	0.947				
Il.Wikifier_Vs._Denote	0.289				

REFERENCES:

[1] Adafre, S. F., & De Rijke, M. (2006, April). Finding similar sentences across multiple languages in wikipedia. *In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics.* pp. 62-69.

[2] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., & Hellmann, S. (2009). DBPedia-A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3), 154-165.

[3] Bunescu, R. C., & Pasca, M. (2006, April). Using Encyclopedic Knowledge for Named entity Disambiguation. *In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics.* Vol. 6, pp. 9-16.

[4] Chi-square: https://statistics.laerd.com/spss-tutorials/chi-square-test-for-association-using-spss-statistics.php. Last accessed September 9, 2014.

[5] Cimiano, P., Handschuh, S., & Staab, S. (2004, May). Towards the self-annotating web. *In Proceedings of the 13th international conference on World Wide Web*. pp. 462-471.

[6] Cornolti, M., Ferragina, P., & Ciaramita, M. (2013, May). A framework for benchmarking entity-annotation systems. *In Proceedings of the 22nd international conference on World Wide Web*. pp. 249-260.

[7] Corpora: http://acube.di.unipi.it/bat-framework Last accessed September 9, 2014.

[8] Cucerzan, S. (2007, June). Large-Scale Named Entity Disambiguation Based on Wikipedia Data. *In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning.* Vol. 7, pp. 708-716.

[9] DBPedia Spotlight Demo: http://DBPedia-spotlight.github.io/demo/. Last accessed September 9, 2014.

[10] Denote Demo: http://www.inextweb.com/denote_demo. Last accessed June 9, 2014

[11] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., & Zien, J. Y. (2003). A case for automated large-scale semantic annotation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1), 115-132.

[12] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., & Zien, J. Y. (2003, May). SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. *In Proceedings of the 12th international conference on World Wide Web* pp. 178-186.

[13] Dingli, A., Ciravegna, F., & Wilks, Y. (2003, October). Automatic semantic annotation using unsupervised information extraction and integration. *In Proceedings of SemAnnot 2003 Workshop.*

[14] Fellbaum, C. (2010). WordNet: An electronic lexical database. 1998. WordNet is available from http://www. cogsci. princeton. edu/wn. Last accessed September 9, 2014.

[15] Ferragina, P., & Scaiella, U. (2012). Fast and Accurate Annotation of Short Texts with Wikipedia Pages. *IEEE software,* 29(1) pp 70-75.

[16] Gabrilovich, E., & Markovitch, S. (2006, July). Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. *In AAAI'06 proceedings of the 21st national conference on Artificial intelligence* Vol. 2, pp. 1301-1306.

[17] Gardner, J. J., & Xiong, L. (2009, November). Automatic link detection: a sequence labeling approach. *In Proceedings of the 18th ACM conference on Information and knowledge management* pp. 1701-1704.

[18] Greaves, M., DAML - DARPA Agent Markup Language http://www.daml.org/. Last accessed September 10, 2014

[19] Hachey, B., Radford, W., Nothman, J., Honnibal, M., & Curran, J. R. (2012). Evaluating entity linking with Wikipedia. *Artificial intelligence,* Vol.194, pp. 130-150.

[20] Handschuh, S., Staab, S., & Ciravegna, F. (2002). S-CREAM—semi-automatic creation of metadata. *In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web* pp. 358-372.

[21] Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., ... & Weikum, G. (2011, July). Robust disambiguation of named entities in text. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing* pp. 782-792.

[22] Illinois Wikifier Demo: http://cogcomp.cs.illinois.edu/demo/wikify/. Last accessed September 9, 2014.

[23] Kogut, P. A., & Holmes III, W. S. (2001, October). AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. *In First International Conference on Knowledge Capture (K-CAP 2001) Workshop on Knowledge Markup and Semantic Annotation.*

[24] Kulkarni, S., Singh, A., Ramakrishnan, G., & Chakrabarti, S. (2009, June). Collective annotation of Wikipedia entities in web text. *In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* pp. 457-466.

[25] Maynard, D. (2003). Multi-source and multilingual information extraction. *Expert Update*, 6(3), 11-16.

[26] Maynard, D., & Greenwood, M. A. (2012). Large Scale Semantic Annotation, Indexing and Search at The National Archives. *In Proceedings of Language resources evolution conference* pp. 3487-3494.

[27] Medelyan, O., Witten, I. H., & Milne, D. (2008, July). Topic indexing with Wikipedia. *In Proceedings of the Association for the Advancement of Artificial Intelligence WikiAI workshop*. pp. 19-24.

[28] Meij, E., Weerkamp, W., & de Rijke, M. (2012, February). Adding semantics to microblog posts. *In Proceedings of the fifth ACM international conference on Web search and data mining*. pp. 563-572.

[29] Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011, September). DBPedia Spotlight: shedding light on the web of documents. *In Proceedings of the 7th International Conference on Semantic Systems*. pp. 1-8.

[30] Mihalcea, R., & Csomai, A. (2007, November). Wikify!: linking documents to encyclopedic knowledge. *In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*. pp. 233-242.

[31] Milne, D., & Witten, I. H. (2008, October). Learning to link with wikipedia. *In Proceedings of the 17th ACM conference on Information and knowledge management.* pp. 509-518.

[32] Mossberg, W. (2001). New Windows XP Feature Can Re-Edit Others' Sites. *The Wall Street Journal*, *23.*

[33] Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., & Goranov, M. (2003). KIM–semantic annotation platform. *In proceedings of the Semantic Web-International Semantic Web Conference 2003* pp. 834-849.

[34] Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., & Kirilov, A. (2004). KIM-a semantic platform for information extraction and retrieval. *Natural language engineering*, 10 (3-4), 375-392.

[35] Ratinov, L., Roth, D., Downey, D., & Anderson, M. (2011, June). Local and global algorithms for disambiguation to wikipedia. *In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*-Volume 1 pp. 1375-1384.

[36] Reeve, L., & Han, H. (2005, March). Survey of semantic annotation platforms. *In Proceedings of the 2005 ACM symposium on Applied computing*. pp. 1634-1638.

[37] Sapkota, K., Aldea, A., Duce, D. A., Younas, M., & Bañares-Alcántara, R. (2011, October). Semantic-art: A framework for semantic annotation of regulatory text. *In Proceedings of the fourth workshop on Exploiting semantic annotations in information retrieval* pp. 23-24.

[38] Shen, W., Wang, J., Luo, P., & Wang, M. (2012, April). Linden: linking named entities with knowledge base via semantic knowledge. *In Proceedings of the 21st international conference on World Wide Web*. pp. 449-458.

[39] Shen, W., Wang, J., Luo, P., & Wang, M. (2012, August). LIEGE:: link entities in web lists with knowledge base. *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. pp. 1424-1432.

[40]SPSS:https://statistics.laerd.com/spss-tutorials/kruskal-wallis-h-test-using-spss-statistics.php. Last accessed September 9, 2014.

[41] Strube, M., & Ponzetto, S. P. (2006, July). WikiRelate! Computing semantic relatedness using Wikipedia. *In proceeding of 21st national conference on Artificial inteligence* Vol. 2, pp. 1419-1424.

[42] Suchanek, F. M., Kasneci, G., & Weikum, G. (2007, May). Yago: a core of semantic knowledge. *In Proceedings of the 16th international conference on World Wide Web*. pp. 697-706.

[43] Suchanek, F. M., Kasneci, G., & Weikum, G. (2008). Yago: A large ontology from Wikipedia and Wordnet. *Journal Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3), 203-217.

[44] T.Berners-Lee, J.Hendler, and O.Lassila. The Semantic Web. *Scientific American*, 1(501), May 2001.

[45] TagMe Demo: http://tagme.di.unipi.it/. Last accessed September 9, 2014.

[46] Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., & Ciravegna, F. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *In Web Semantics: science, services and agents on the World Wide Web*, 4(1), 14-28.

[47] Vargas-Vera, M., Moreale, E., Stutt, A., Motta, E., & Ciravegna, F. (2007). MnM: semi-automatic ontology population from text. *In Book: Ontologies.* pp. 373-402.

[48] Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., & Ciravegna, F. (2002). MnM: Ontology driven semi-automatic and automatic support for semantic markup. *In proceeding Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web.* pp. 379-391.

[49] Weaver, G., Strickland, B., & Crane, G. (2006, June). Quantifying the accuracy of relational statements in wikipedia: a methodology. *In proceeding of Joint Conference on Digital Libraries.* Vol. 6, pp. 358-358.

[50] Wikimeta Demo: http://www.wikimeta.com/index.html. Last accessed September 9, 2014.

[51] Wikipedia Miner Demo: http://wikipedia-miner.cms.waikato.ac.nz/demos/annotate/. Last accessed September 9, 2014.