TermPedia for Interactive Document Enrichment

Using Technical Terms to Provide Relevant Contextual Information

Proscovia Olango
Email: p.olango@rug.nl

Gerwin Kramer
Email: G.Kramer.1@student.rug.nl

Gosse Bouma
Email: g.bouma@rug.nl

***Abstract*—TermPedia is a human language technology (HLT) application for document enrichment that automatically provides definitions for technical terms (TTs). A technical term (TT) may hinder document comprehension if it is introduced without any definition or explanation. In some cases when a term is defined, the definition may contain additional technical terms that instigate a similar problem. This is why we investigated a possibility of providing contextually relevant information for the technical term by linking it to an encyclopedia. In this way, additional information relating to the technical terms shall be readily available and hopefully make documents more comprehensible.**

I. Introduction

Document content comprehension is an important component of beneficial reading without which reading would be futile. Content comprehension may be defined as the extent to which document text is easily understood by a reader. Text that can be easily understood is said to be accessible and there are a number of factors that influence text accessibility. Vocabulary was listed in [1] as the first factor influencing text difficulty and accessibility besides sentence structure, length, elaboration and others. Replacing difficult words with easier ones may not simplify text but rather deny readers an opportunity to expand their vocabulary. An additional problem is that both the original technical vocabulary and the replacement may be ambiguous, which make automatic replacement a difficult task.

When proficient adult readers struggle with technical expository text on unfamiliar arcane topics, reading is slowed to a near halt and comprehension is seriously compromised [2]. Effort and time is needed since a reader may consult external information resources in order to comprehend a piece of text. Fortunately we can employ the technique of document enrichment to improve understanding of documents by providing additional contextual information about TTs. The notion of document enrichment is conceived to reduce the effort and time needed in acquiring additional information by defining and linking TTs to contextually relevant encyclopedic knowledge. To facilitate document content comprehension we propose TermPedia, a system that recognizes a technical vocabulary and links it to a relevant Wikipedia article. This integrates explanations of the vocabulary into the document and hopefully this shall provide adequate information for content comprehension.

We define a technical term (TT) as a newly introduced or uncommon word or combination of words within a particular knowledge domain. TT also refers to a common word or combination of words used with a special meaning in the context of a particular document. For easy reference and writing, acronyms and abbreviations are also considered as TTs in this paper. The example below depicts a sentence[1] which may be difficult to comprehend for a reader who is not conversant with the medical knowledge domain. TTs have been marked in bold.

> **Epidural hematoma** (**EDH**) is a rapidly accumulating **hematoma** between the **dura mater** and the **cranium.**

The example among others presents the abbreviation or acronym "EDH", it is clear that this abbreviation refers to *epidural hematoma*, but what exactly does this mean? Although the TT *epidural hematoma* is explained in the sentence example, the explanation still makes no sense if the reader does not understand the TTs it contains. We therefore see that sometimes when a TT is explained in a document, additional contextual information is necessary for the reader to understand it.

A. Document Enrichment

Document enrichment employs techniques of information extraction and text mining for integrating knowledge into existing text. A major technique in document enrichment is that of generating interactive text. One way to generate interactive text is by creating hypertext. This involves the process of tagging text with anchors that lead to external or internal information resources of a documen [3]. In order to accomplish the task of document enrichment, we used TTs as anchors. The anchors are important as hypertext links to additional encyclopedia information. The experiment assimilated work done by [4] who try to link educational materials to encyclopedic knowledge. Encyclopedias are a large source of authentic information but to make good use of this information, a user has to switch between interfaces, whether on the World Wide Web or while using hard copies of these references. We hope to minimize the time and distance between available information and text by integrating Wikipedia knowledge into documents.

Wikipedia is a free-content online encyclopedia resultant from continuous collaborative effort of volunteer contributors. Although a few critics question the credibility and coverage of Wikipedia, in the year 2005 a special report on science

[1] See *http://en.wikipedia.org/wiki/Head_injury* for this sentence.

articles indicated that Wikipedia is similar to *Encyclopedia Britannica* in both coverage and accuracy [5]. Questions about the accuracy of Wikipedia arise because many of the contributors are not accredited authors. Contrarily this is seen as an advantage by the co-founders of Wikipedia as they anticipate that any error noticed on the content pages shall be instantly corrected by the people who notice them.

We experimented with document enrichment for text from the medical domain. The reason for choosing this domain is that on the one hand it is a field which uses highly specialized terminology, and on the other hand, non-experts from time to time have the need to consult medical texts. We believe that the results obtained in our experiments do carry over to other domains.

The rest of this paper is organized as follows; section II talks about other research which has been done that is related to TermPedia, section III describes the methods used in TermPedia, section IV reports a pilot experiment that was carried out using medical articles from Wikipedia, section V describes a web-based interface that was developed to make TermPedia accessible to users, and section VI provides concluding remarks and suggests future work.

II. Related work

This section discusses work done on term extraction and automatic hypertext generation, which are essential techniques for document enrichment.

Term Extraction: [6] focus on the use of existing terms from glossaries, thesaurus, or ontologies to extract new terms from a domain specific text. Their baseline system combines a linguistic pattern for extracting candidate noun phrases with a statistical method for ranking candidate phrases according to their association strength in a domain specific corpus. They developed a method for ranking candidate terms, extracted from Dutch medical corpora, with the help of the Unified Medical Language Systems (UMLS) as an external knowledge source. [6] concentrated on extraction of phrasal terms and their method combines frequency of occurrence of candidate terms in a corpus with information on how much the candidate terms overlap with existing multilingual terms. Noting that it is only phrasal terms that were considered, it could be a good idea to provide their definitions since the terms are extracted from existing knowledge bases. It would be interesting to extend the term extraction for other domains of knowledge by using other external knowledge sources in addition to UMLS.

Other methods used in term extraction include tf-idf weight (term frequency-inverse document frequency), term co-occurrence, and concept identification using Wikipedia. The last method considers Wikipedia article page titles as terms and these are in turn used to recognize terms in plain text documents. [7] made an evaluation of these methods and reported that the Wikipedia technique was significantly more effective than the other techniques. For this reason, we were motivated to use Wikipedia article titles and links embedded within the articles as a baseline for technical term extraction.

Automatic Link Generation (ALG): ALG involves techniques like automatic term definition [8] and word sense disambiguation (WSD). WSD is the process of accurately and automatically identifying the sense of a word as used in context. Many techniques have been used in this process including the use of machine readable dictionaries. In the latter approach a word sense is guessed by counting overlaps between dictionary definitions of various word senses and the context where a word appears in text. Michael Lesk who is known for introducing the Lesk algorithm which is a classical algorithm for WSD, reports that his system disambiguates word sense with an accuracy rage between 50% to 70%, on text from *Pride and Prejudice* and selected papers of the Associated Press [9]. Notice that only single words are disambiguated in Lesk's paper and yet the majority of technical terms are compound nouns or word phrases. [10], mention that "85% of domain specific terms are said to be compound nouns". Therefore as we apply the technique of WSD for disambiguating the meaning of technical terms, we take into consideration that most of these terms are compound nouns or a combination of word phrases.

Using Knowledge Bases for Document Enrichment: Recent years have seen an enthusiastic growth of research in the area of using existing knowledge bases to enrich documents [e.g 11; 12]. This is a logical development that arises from the need to comprehend technical documents. A motivation for our project is the work done by [11] who introduce the use of Wikipedia as a resource for automatic keyword (technical term) extraction and word sense disambiguation. The two methods were combined into a system called Wikify! which automatically performs the annotation task following Wikipedia guidelines. If a document is given as input to Wikify!, the system has the ability to identify the important terms in the text and link them to corresponding Wikipedia pages. These links provide users with quick access to contextually related information. We generalized this process as document enrichment however, Mihalcea and Csomai do not take into consideration the importance of providing definitions for the terms identified. We find it important to provide definitions for the terms because in some cases these definitions are sufficient to quench a reader's thirst for comprehension. A Turing test evaluation showed that Wikify! generates Wikipedia annotations that are hardly distinguishable from the ones that are human-generated.

III. Description of TermPedia

TT prediction, and automatic hypertext generation are the two distinct approaches used in TermPedia for document enrichment. We identified TTs and provided definitions for these. Keeping in mind that some terms are ambiguous, we performed sense disambiguation in order to provide a contextually relevant definition for each term. During the process of linking TTs to their definitions, hypertext was generated. This was an important step for integrating available information into documents. On the other hand, we were not only interested in the definition and explanation of TTs because we are

doubtful that this will totally satisfy the reader's quest for content comprehension. Therefore, we also linked the terms to appropriate Wikipedia articles depending on their contextual disambiguation.

A. Technical Terms Prediction

TT prediction explored four supervised methods for automatic term recognition. The four methods explored were: (1) blindfold term prediction (BTP); (2) longest-string-based term prediction (LTP); (3) sense-based term prediction (STP) and (4) frequency-based term prediction (FTP). (1) and (2) methods did not consider TT disambiguation.

1) Blindfold Term prediction (BTP): The BTP method used a string look-up technique which marks all possible strings in a document that match a term from an existing terms list as a TT. The Last in First Out (LIFO) principle borrowed from the top of a stack data structure [See 13] was used for each sentence within the plain text. We believe that a sentence contains a stack of TTs and the first term to be recognized in that sentence is then predicted, there is also a first come first serve idea for the terms. This method is expensive because each line of running text is matched against all the TTs in an existing terms list.

2) Longest-String-Based Term prediction (LTP): The difference between BTP and LTP is that in LTP the existing terms list was sorted in such a way that the terms having the longest length of characters were stored at a queue head. LTP was thus able to predict the longest string in running text that matched a term from the existing terms list as a TT. This criterion was adopted to avoid splitting up terms that are made of compound nouns or noun phrases. For example, if the term *Epidural hematoma* is not extracted first, it may be split into *Epidural* and *hematoma*. The first come first serve idea here is biased towards the longest string in a sentence that can be predicted as a TT. Like BTP, LTP did not take into consideration ambiguity of term senses in relation to context.

3) Frequency-Based Term prediction (FTP): For the FTP approach we developed a criterion based on the keyword ranking method called *Keyphraseness*, which was presented by [11]. In this approach all possible n-grams in a document that are present in an existing list of terms are identified and ranked according to their likelihood of being selected as a TT. If a term is most of the time selected as a TT among its total number of occurrence, it is most likely that it will again be selected in a new document as a TT. Therefore the probability (*P*) that a term *(X)* is selected as a TT in a new document is calculated as the total number of documents where the term was already selected as a TT *(count($D_{TT,X}$))* divided by the total number of documents where the term appeared *(count(D_X))*.

$$P(TT|X) \approx \frac{count(D_{TT,X})}{count(D_X)} \quad (1)$$

The counts were generated from all the articles in a 2006 Wikipedia dump. Given a list of TTs in a document, we select the top 10% although [11] note that on average, 6% of the TTs in a Wikipedia article are actually linked to another page. The FTP method also performs TT sense disambiguation by picking the most frequent term sense.

4) Sense-Based Term prediction (STP): A database containing TTs and their definitions was created to facilitate the STP method. To create the database anchor texts from the 2006 XML Wikipedia dump were considered as TTs regardless of the number of n-grams they contained. An anchor text is a string of characters that occur between the "$< a >$" tag of a hypertext markup language (html). XQuery, a language designed to query a collection of XML documents was used to extract the anchor texts and the Wikipedia articles to which they were linked. XQuery uses XPath to traverse the XML version of Wikipedia for retrieving this information. XPath is a language for finding information in an XML document.[2]

TT definition was done by carrying out a target look-up for the term. Taking a close look at Wikipedia pages we noticed that the first paragraph is often a definition of the title of that page. Conventionally a Wikipedia anchor text (TT) is linked to a Wikipedia page (the target), therefore the definition of the TT was taken as the first paragraph of the target page. We trust that a TT is linked to a contextually relevant Wikipedia page by the contributing authors.

TABLE I
SAMPLE LIST OF THE MOST FREQUENT WORDS USED IN WIKIPEDIA

Position	Common Word	Frequency
1	the	68730054
2	of	37253050
.	.	.
.	.	.
50	can	1149892
51	only	1142748
.	.	.
.	.	.
99	york	528892
100	day	527048

STP took into consideration the contextual meaning of a term before it was predicted as a TT by overlapping the words in the paragraph where the term occurs with the words in the term's definition. If there is high overlap of words between the two sets of text, then we assumed that their context are similar. An intersection computation algorithm was used to disambiguate contextual meaning of the TTs. Given the two paragraphs, we determined which words are common to both paragraphs (intersection). Each paragraph was tokenized and sorted to have unduplicated items. A list of the most common words used in Wikipedia was created to form a stop words list for the method. The stop words list was created by a frequency count of all the words that occurred in the entire Wikipedia dump of 2006 using a plain text version. The most frequent 100 words were then used as a list of stop words. Table I shows 9 of the most frequent 100 words in the list and their frequencies.

[2]See, *http://w3schools.com/xpath/default.asp*. Referenced on 9 Jan., 2009.

The stop words were removed from the tokenized paragraphs to form a pair of word lists P_1, and P_2 that contain only the important words within the paragraphs. P_1 was a list of words derived from the paragraph containing the term definition and P_2 was the word list derived from the paragraph where the term occurred in a new text document. To find the similarity rank between the two paragraphs ($SR_{P_1 and\ P_2}$), a count of the words in their intersection *(count*$(P_1 \cap P_2)$*)* was determined. The least threshold rank was set to 2. We expected STP to have the best recall since it was designed to predict TTs after disambiguating their senses in relation to context.

$$SR_{P_1 and\ P_2} \approx count(P_1 \cap P_2) \geq 2 \quad (2)$$

B. Automatic Hypertext Generation

A database containing TTs and Wikipedia articles to which the terms are linked was created to facilitate the process of automatic hypertext generation. An HTML $< a >$ (anchor) tag is inserted around a TT so as to create the hypertext. The Uniform Resource Locator (URL) of the Wikipedia article to which the TT is linked served as an hypertext reference (href) attribute of the $< a >$ tag. For example if the TT *Epidural hematoma* is predicted by the system, the term will become a hypertext after the following transformation:

```
<a href="http://en.wikipedia.org/wiki/
  Epidural_hematoma">Epidural hematoma</a>
```

In some cases a term was linked to more than one Wikipedia article (ambiguous TT). For such terms a criterion similar to that described in III-A4 was used and the article with the highest overlap rank was given as the term's target. Hypertext generation completes the document enrichment process because once users click on the hypertext, they are presented with a Wikipedia article that defines and explains the TT in question. For a user-friendly environment, a Java script hover function was provided to pop-up a window with only the definition of the TT (anchor text) that includes a link to more information from Wikipedia. If a user is not satisfied with the definition, he can conveniently link to the encyclopedia for more explanation. The final system was an application of integrated techniques including TT prediction, automatic term definition, TT sense disambiguation, and automatic hypertext generation, which resulted into a system of document enrichment. See section V for the user interface description.

C. Challenges

The biggest problem of this project is that many TTs have several senses because for any given term there could be an ambiguity of how it's sense is interpreted in relation to the context in which it appears. The challenge is therefore to develop a competent disambiguation engine that is able to predict the accurate interpretation of a term in context.

Since we used an existing terms list, it is possible that the list does not have all the TTs that occur in a specific document. This challenge could be reduced by using multiple term lists.

Inter-annotation disagreements is another challenge that this research faces. Experiments have shown that given the same piece of text, different humans annotate different TTs. Although this disagreement is controlled by different reading levels of users as demonstrated by [12] and [4], it is likely that certain Wikipedia contributors overlook TTs in particular articles. Additional information will not be provided for the overlooked terms and these terms may still hinder the understanding of a document by someone at a lower reading level.

IV. Pilot Experiments and Evaluation

We used the entire English UVA XML Wikipedia dump of November 2006, which contained more than 5,000,000 articles with over 3,000,000 non-redirect articles as our main corpus. The XML corpus was created by the Information and Language Processing Systems (ILPS) department, Informatics Institute, University of Amsterdam.[3] This XML version of Wikipedia was developed to serve as a multi-lingual text collection for experiments in Information Retrieval and Natural Language Processing, in the context of Cross-Language Evaluation Forum (CLEF). We use the corpus for extraction of TTs which is in line with the information retrieval intentions for its creation, but only for the English language. We also use the English XML Wikipedia dump of January 2008 which was compiled by Wikimedia Foundation Incorporation.[4] From this dump, 4,347 articles that belong to the medical category are considered. These two sets of data come with all the Wikipedia XML formatting information which we removed to generate a clean text version of the Wikipedia articles.

Articles from the medical category of Wikipedia were collected by John Kizito[5], from which we collected medical page ids. By using the medical page ids, we generated a medical corpus from the UVA XML Wikipedia dump of March 2006. The medical corpus was created by extracting Wikipedia pages with the same ids as those in Kizito's medical corpus from the 2006 dump. A total of 1,166 articles constitute the medical sub-corpus that we used to train and evaluate TermPedia.

The main aim of the experiment is to link terms existing within a plain text document to a contextually relevant Wikipedia page in reference to the terms' context. This experiment was inspired by the work done by [11], who link documents to Wikipedia with the help of anchor text. For more information about anchor text, please see section III-A4 of this paper. Therefore we carried out a supervised experiment to predict which terms in a Wikipedia lemma get tagged with links to other Wikipedia pages.

A. Collecting Medical Anchor Text

From the Wikipedia dump of March 2006 that consisted of 1,166 articles, we collected a total of 30,528 anchor texts and 26,440 page-titles. 1,955 of the 30,528 anchor texts were ambiguous, indication that 6.4% of the TTs in the existing term list were ambiguous. Each anchor text was treated as a medical technical term. We use the anchor text to refer to term lemmas

[3]See, *http://ilps.science.uva.nl/WikiXML/*. Referenced on 9 Jan., 2009

[4]See, *http://download.wikimedia.org/enwiki/*. Referenced on 7 Feb., 2009

[5]John Kizito (2008). PhD Student, Faculty of Computing and Information Technology, Makerere University.

TABLE II
OUTPUT OF COLLECTED MEDICAL ANCHOR TEXTS AND TARGETS

Page-Id	Page-Name	Page-Title (Target)	Anchor Text
353792	Folk_medicine	Herb	herbs
310484	Patent_medicine	Herb	herbal
2142761	Leukapheresis	Blood plasma	Plasma

because they reflect the exact form in which terms are written in documents and this was also useful for dealing with spelling variances. For each anchor text we collected information about the Wikipedia page-id, page-name, and page-title to which that anchor refers. The anchor text collection consisted of single words, groups of words, abbreviations, and acronyms. The page-titles collected alongside these anchor texts tell us which Wikipedia page the anchor text is linked to.

TableII shows that the page with id *2142761* on *"Leukapheresis"* has a link to the Wikipedia page *"Blood plasma"* with anchor text *"Plasma"*.

B. Document Enrichment Using Anchor Text

By now, we have a list of medical anchor texts (or term lemmas) and page-titles of the Wikipedia pages to which the anchor texts are to be linked. This makes it possible for us to enrich plain text documents with information from Wikipedia by automatically generating hypertext using the existing list of anchor texts. Some anchor text refers to more than just one Wikipedia page-title. For example, the anchor text *Avicenna* refers to both *Avicenna* and *Avicenna (crater)* page-titles, therefore a disambiguation task had to be performed. See section III-B above.

1) Finding Anchor Text in Plain Text: We retrieved all possible Wikipedia page targets for each anchor text and each page-title. For example, our challenge during the disambiguation process is to make sure that, given a sentence like *Avicenna also introduced medical herbs*, the anchor text *Avicenna* is linked to the Wikipedia page-title *Avicenna* and not *Avicenna(crater)*. For this example there were three TTs recognized by the methods BTP, LTP and STP as shown in table III.

TABLE III
PREDICTION AND LINKING TT TO WIKIPEDIA BY TERMPEDIA

Recognized TT		Recognized Target (Name of Wikipedia article)
i.	Avicenna	Avicenna
ii.	medical	Medical_care
iii.	herbs	Herbalism

2) Results and Evaluation: We evaluated the performance of the four methods used for recognizing TTs from text and linking them to Wikipedia articles. The original Wikipedia pages came in eight zipped files with the file names *wikipedia-en**0**.txt.gz* to *wikipedia-en**7**.txt.gz*. A random number of 151 Medical articles were selected from the *wikipedia-en**6**.txt.gz* and *wikipedia-en**7**.txt.gz* zipped files for the evaluation purpose. To evaluate the methods we extracted all automatically added links they generated and compared these to the links that existed in the original Wikipedia articles.

Evaluation of term recognition and automatic link generation for the four TermPedia methods was done by calculating precision, recall and F-score for each method. F-score in particular measures a system's accuracy and reaches its best value at 1 and worst at 0. The overall F-scores which are presented in Tables IV and V below are the F-scores of the average precision and recall for each system. In the case of term recognition, precision is the number of correctly recognized terms divided by the number of all terms that were recognized by the methods and recall is the number of correctly recognized terms divided by the number of terms that exist in the original Wikipedia articles.

TABLE IV
OVERALL EVALUATION OF TECHNICAL TERM RECOGNITION

Method	#docs	Precision	Recall	F-score
BTP	151	0.186	0.679	0.292
LTP	151	0.237	0.864	0.372
FTP	151	0.332	0.742	0.458
STP	151	0.215	0.808	0.340

Similarly, for automatic link generation, precision is the number of links correctly generated by the methods divided by the number of all links that were generated by the methods. Recall is the number of links that were correctly generated by the methods divided by the number of links that exist in the original Wikipedia articles.

LTP method had the best recall of 86.40%. We suspect that the longest match look-up criteria is partly responsible for the 14% lost terms because a term can only be recognized once and if it is seen in the longest string then it will not be seen again.

Interestingly, all the four methods have poor precision. Although FTP has the best precision of 33.20% it reflects a much lower standard result than that presented by [11] in their method of *keyphraseness* which had a precision of 53.37%. Unfortunately we cannot compare these results literally because [11] use the entire Wikipedia and we only consider articles from the medical category.

TABLE V
OVERALL EVALUATION OF LINK (TT AND TARGET) PREDICTION

Method	#docs	Precision	Recall	F-score
BTP	151	0.153	0.586	0.243
LTP	151	0.204	0.771	0.322
FTP	151	0.309	0.689	0.427
STP	151	0.195	0.762	0.311

Table V gives results for anchor text recognition and target prediction. This is a harder task so we expected lower scores, which the figures above confirm. A 77.1% recall in the case of LTP is low for a system that uses a supervised method,

because what this method does is simply to assign links to terms that have been linked before. The method does not take into consideration the frequency of the term or its contextual sense. Perhaps this may be a reason for its low recall. The best system for both term prediction and automatic link generation is revealed as the FTP system that excels with F-scores of 45.8% and 42.7% for the two tasks respectively.

FTP outperforms the STP method because the overlap ranks are very low for the term context and term definition paragraphs. We set a cutoff as low as 2 for the overlap rank and still ended up with these results. It is also possible that a paragraph is too long for generating an accurate word overlap. May it would be better to take a few words to the left and right of where the terms occur and overlap these instead.

3) Discussion of Results: By taking a closer look at the overall evaluation results in tables IV and V, it was noticed that TermPedia methods predicted more TTs for the articles than the ones that were indicated by contributing authors. This was the main reason for the general poor precision results by the methods regarding TTs prediction.

In table VI we can see that for 10 randomly selected Wikipedia articles, a total of 635 new terms were predicted by the STP method as compared to a total of 155 terms that originally existed in these articles. The total number of new predicted TTs is well over 50% consequently producing low precision. The "Overlap" column of the table presents the number of TTs that exist in an original Wikipedia article that were predicted by STP. For this experiment, these are the good terms because it means that STP was able to accurately predict terms that were indicated by the contributing authors. STP was able to predict all the TTs that existed in 4 of the 10 randomly selected articles as shown by zero values in their missed column.

The "Missed" column in table VI represent the number of terms that STP method was not able to predict although these terms were indicated in the original articles (gold standard) as TTs by the contributing authors. Fortunately, STP fails to predict just a few TTs that already exist in the gold standard. For this reason, the method has an overall average recall of 90.4% in predicting TTs for the 10 randomly selected documents.

Key for table VI

- Article Size (Bytes): Size of random Wikipedia article in bytes.
- Gold: Total number of TTs that exist in an original Wikipedia article.
- Overlap: Total number of TTs that exist in an original Wikipedia article that were predicted by STP.
- Missed: Total number of TTs that exist in an original Wikipedia article that were not predicted by STP.
- New TTs: Total number of TTs that were predicted by STP but did not exist in an original Wikipedia article.
- All TTs: Total number of TTs that were predicted by STP in the text version of a Wikipedia article.
- P=Precision, R=Recall, and F=F-score.

The low precision scores may not be problematic because the major objective of the project is to identify TTs in text. It is likely that large number of newly predicted TTs point to related Wikipedia articles. If so, the system merely adds new relevant links to a document. The question as to what percentage of text in a document should be marked as TTs can be controlled as long as the TTs are accurately predicted. See section V.

TABLE VI
STATISTICS OF TTS PREDICTED BY STP AGAINST GOLD STANDARD

Article Size (Bytes)	Gold	Overlap	Missed	New TTs	All TTs	P (Overlap) (All TTs)	R (Overlap) (Gold)	F (2RP) (P+R)
560	5	5	0	6	11	0.455	1.000	0.625
1,235	6	4	2	29	33	0.121	0.667	0.205
1,773	22	19	3	31	50	0.380	0.864	0.528
2,123	7	7	0	34	41	0.171	1.000	0.292
2,534	11	11	0	76	85	0.129	1.000	0.229
2,924	4	4	0	42	46	0.087	1.000	0.160
3,210	31	28	3	60	88	0.318	0.903	0.471
3,438	20	17	3	58	75	0.227	0.850	0.358
7,741	17	16	1	141	157	0.102	0.941	0.184
17,173	32	26	6	158	184	0.141	0.812	0.241
Average	**15.5**	**13.7**	**1.8**	**63.5**	**77.0**	**0.213**	**0.904**	**0.329**

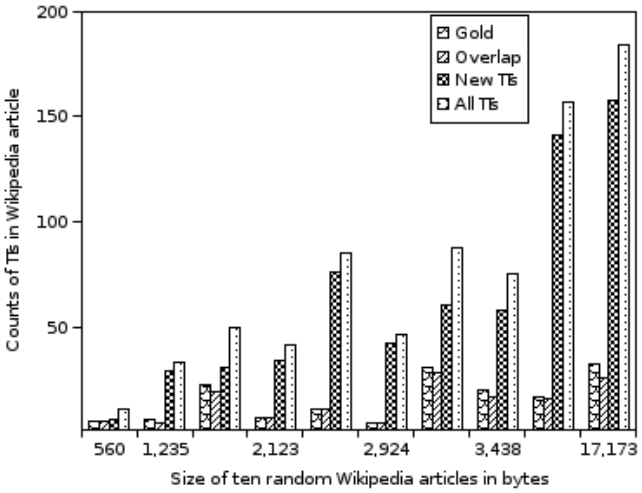

Fig. 1. Counts of TTs found in gold standard compared to those predicted by STP method

Figure 1 presents a bar chart plot of the performance of STP against gold standard for the 10 randomly selected documents that appear in table VI. For the plot TTs counts in each random Wikipedia article were plotted on the y-axis and the size of each Wikipedia article in bytes was plotted on the x-axis. The height of bars clearly show that the number of TTs that could be predicted by STP method are directly proportional to the size of the articles. This proportionality provided confidence that the method was executing correctly because it is logical to find more TTs in an article of bigger size compared to an article of smaller size. The biggest article shown in figure 1 had a size of 17,173 bytes and for this article STP predicted a

total 184 TTs as shown by the "All TTs" column. Comparably STP predicted only a total of 11 TTs from the smallest article of size 560 bytes.

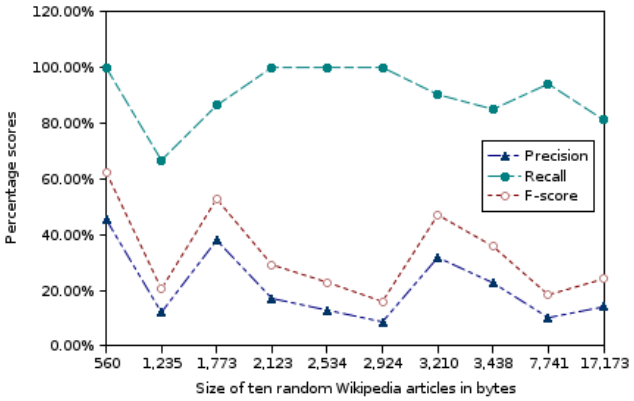

Fig. 2. Percentage evaluation scores of STP against gold standard

Figure 2 clearly shows that the STP had a very good recall of predicting TTs in these 10 articles. The recall line plot is well over 50% with the lowest recall at 66.7% and the highest recall at 100%. From this figure we can also see that the low values of precision greatly affect the value of F-scores. The F-score plot presents a perfect transpose of the precision plot with a constant average distance of 20% in the direction of F-score. The lowest F-score value of how well STP can predict TTs in the 10 randomly selected documents was 16% and the highest F-score value was 62.5%. In order to improve the F-score values, the precision values need to be improved. This can be done by allowing the system to predict fewer TTs than it is currently predicting. One way to do this is by using a percentage threshold of the terms that can be predicted by the system in relation to the articles' size. For the FTP method n-grams generation may be affecting the performance of the method because during the generation of these n-grams, text is tokenized at the space character. If n-grams are not carefully generated, the string look-up process of FTP will not be able to match the generated n-grams with the existing list of TTs although some of the n-grams actually present a TT. This problem is mainly caused by n-grams that include punctuation marks in the running text. Although we tried to solve this problem by improving the robustness of the tokennizer, this solution is not universal because in some cases the punctuation marks are needed.

V. User Interface Description

As our goal is to make electronic documents on the web more easily accessible by providing contextual information, we developed a web-application to demonstrate our work. A screenshot of the web-application is given in figure 3.

The web-application[6] allows the user to enter a URL of a page that he wants to have annotated, or to upload a document. Documents or pages are assumed to be HTML. Pdf-documents can also be processed, but these are first converted to HTML, and the output of the system is also HTML. The system first turns a page into plain text. Next, the text is processed by the FTP document enrichment algorithm (i.e. the approach that currently achieves the highest f-score). The output of the algorithm is, among others, a list of anchor texts and corresponding targets.

This list is taken as input for a module that takes the original html, and adds new hypertext links (pointing to relevant wikipedia pages) to it. Furthermore, each hypertext has the added (javascript) functionality, that the definition of the hypertext (i.e. the first paragraph of the corresponding wikipedia page) is shown in a pop-up window, as soon as the user moves the mouse over the text. Definitions are retrieved in real-time from the current version of Wikipedia with an excellent speed. In addition, all existing links in the page are turned into links which point to the system, so that any new page accessed by the user is also automatically enriched.

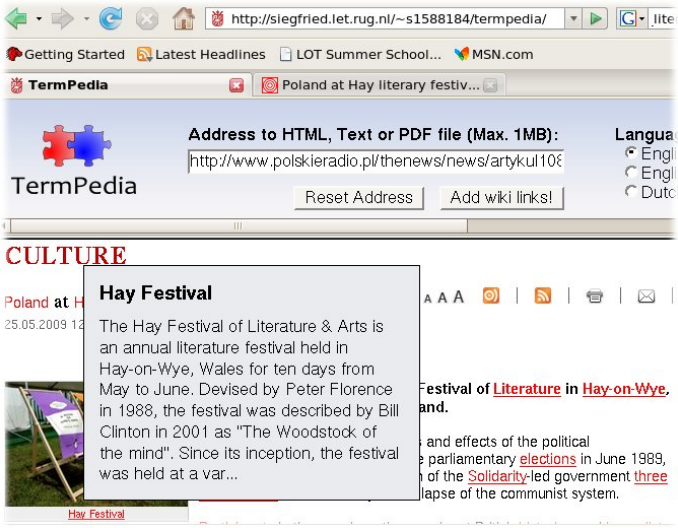

Fig. 3. TermPedia demonstrator

A link density feature is contained in the web-application to allow users decide what percentage of text in a document should be marked as hypertext. The user is also provided with a feature for choosing the colour in which the hypertext should be displayed.

VI. Conclusion and Future Work

Conclusion: We have indicated that it is possible to use TTs and available encyclopedic knowledge to automatically enrich documents by integrating HLT techniques. The results of this resarch could be applied in e-learning. Unfortunately this may not be very feasible for a developing country like Uganda considering expenses related to ICT facilities and the knowledge for using and maintaining them [14]. Fortunately, the proposed project develops a system that does not depend entirely on the Internet. The semantic web could also use the automatic hypertext generation feature of our document

[6]On-line at http://siegfried.let.rug.nl/~s1588184/termpedia/

enrichment process since this results in anchored text that is linked to web pages. Knowledge processing and information extraction applications could use the term extraction feature of this project for improved annotations.

Future Work: The biggest bottle-neck of this research seems to be accurate TT prediction. If we think of this as a machine learning (ML) problem then for each string in the text that has been used as a TT, we could collect features like its frequency, its position, and words that preceded and follow it in an article. These features can then be used in a ML method to improve the general performance of the system in term prediction, thereby also improving the automatic link generation process of the system. A robust way of generating n-grams is also needed, so that punctuation can be dealt with more accurately. The term prediction criteria could be remodeled so that it does not only depend on string look-up.

Other future work could include evaluating TermPedia on medical data outside Wikipedia and integrating the LTP and FTP methods with the aim of improving the application's precision. The existing terms list could be extended by using medical terms from UMLS and eventually a user survey may be carried out.

An important issue of the TermPedia system is user-friendliness, therefore the current user interface shall continually be developed to allow a user-friendly environment. In addition all resultant modules from the system shall be made available as free-ware for interested NLP researches to incorporate into their works whenever necessary and applicable.

REFERENCES

[1] M. F. Graves and B. B. Graves, *Scaffolding Reading Experiences: Designs for Student Success*, 2nd ed., 2003.

[2] D. S. McNamara, Ed., *Reading Comprehension Strategies: Theories, Intervensions, and Technologies*. Routledge, 2007.

[3] J. Domingue, N. Franes, E. Motta, S. B. Shum, M. Vargas-Vera, and Y. Kalfoglou, "Supporting ontology-driven document enrichment within communities of practice," in *The 1st International Conference on Knowledge Capture (K-Cap) proceedings*, 2001. [Online]. Available: http://eprints.aktors.org/12/01/kcap01_john_final.pdf

[4] A. Csomai and R. Mihalcea, "Linking educational materials to encyclopedic knowledge," 2007, unpublished research paper. [Online]. Available: http://www.cse.unt.edu/~rada/papers/csomai.aied07.pdf

[5] J. Giles, "Internet encyclopedias go head to head," *Nature*, vol. Vol. 438, pp. 900–901, December 2005, published online 14 December 2005. [Online]. Available: http://www.nature.com/nature/journal/v438/n7070/full/438900a.html

[6] I. Fahmi, G. Bouma, and L. V. der Plas, "Learning to identify definitions using syntactic features," in *EACL 2006 Workshop on Learning Structured Information in Natural Language Application*, 2007, pp. 64–71. [Online]. Available: http://acl.ldc.upenn.edu/W/W06/W06-2609.pdf

[7] M. S. Jones, "An evaluation of different term extraction techniques for searching ontologies," 2007, unpublished research paper. [Online]. Available: http://portfolio.ecs.soton.ac.uk/19/

[8] R. Torralbo, E. Alfonseca, A. Moreno-Sandoval, and J. M. Guirao, "Automatic generation of term definitions using multidocument summarisation from the web," in *RANLP '05 Workshop on Crossing Barriers in Text Summarization Research proceedings*, 2005. [Online]. Available: http://elvira.lllf.uam.es/ESP/publicaciones/RANLP05resumenes.pdf

[9] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *SIGDOC '86: Proceedings of the 5th annual international conference on Systems documentation*. New Yor, USA: ACM, 1986, pp. 24–26.

[10] H. Nakagawa and T. Moris, "A simple but powerful automatic term extraction method," 2000, unpublished research paper. [Online]. Available: http://www.aclweb.org/anthology-new///W/W02/W02-1407.pdf

[11] M. Rada and C. Andras, "Wikify!: linking documents to encyclopedic knowledge," in *CIKM '07: Proceedings of the sixteenth ACM conference on information and knowledge management*. New York, USA: ACM, 2007, pp. 233–242.

[12] N. Elhadad, "Comprehending technical texts: Predicting and defining unfamiliar terms," in *AMIA Annual Symposium proceedings*, 2006, pp. 239–243. [Online]. Available: http://www.dbmi.columbia.edu/noemie/papers/amia06.pdf

[13] "Data structures and algorithms," 1996, prepared by Erhan Erdem, June. [Online]. Available: http://www.cmpe.boun.edu.tr/~akin/cmpe223/chap2.htm

[14] G. Farrell and I. Shafika, *Survey of ICT and Education in Africa: A Summary Report, Based on 53 Country Surveys*. infoDev/World Bank, Washington, DC, 2007. [Online]. Available: http://www.infodev.org/en/Publication.353.html