RealText$_{cs}$ - Corpus Based Domain Independent Content Selection Model

Rivindu Perera, Parma Nand
School of Computing and Mathematical Sciences
Auckland University of Technology
Auckland 1010, New Zealand
{*rivindu.perera, parma.nand*}*@aut.ac.nz*

***Abstract*—Content selection is a highly domain dependent task responsible for retrieving relevant information from a knowledge source using a given communicative goal. This paper presents a domain independent content selection model using keywords as communicative goal. We employ DBpedia triple store as our knowledge source and triples are selected based on weights assigned to each triple. The calculation of the weights is carried out through log likelihood distance between a domain corpus and a general reference corpus. The method was evaluated using keywords extracted from QALD dataset and the performance was compared with cross entropy based statistical content selection. The evaluation results showed that the proposed method can perform 32% better than cross entropy based statistical content selection.**

***Keywords*-Content Selection; Natural Language Generation; Semantic Web; Natural Language Processing**

I. Introduction

Content selection is the task responsible for choosing relevant information that should be conveyed in computer generated text given a particular concept as communicative goal [1]. For instance, the content selection framework used in a system to acquire information to build a biography of a person, will use the name of the person as a communicative goal and will select the information needed (e.g., date of birth, education, etc.) from a knowledge source. The knowledge source is the information repository from where content selection framework can acquire relevant information. In the previous example, a knowledge base with personal information can act as a knowledge source. However, content selection is considered to be an extremely domain dependent task. Due to this high level of domain dependency, designing a domain independent content selection framework with reasonable accuracy is considered to be a challenging task.

The work described in this paper is based on the content selection framework (RealText$_{cs}$) that forms part of a larger project for Natural Language Generation (NLG) in open domain Question Answering (QA). Open domain QA is meant to work in different domains, hence an important goal in such a framework is domain independency. In open domain frameworks, there is no opportunity to specify predetermined rules to select content because they would not work with a change in domain. Existing content selection models involve a general set of predetermined rules or similar preprocessing steps before content selection. For instance, Demir et al. [2] use predetermined prepositions relevant for the domain. A model presented by Duboue and McKeown [3], employs an initial step to determine rules which can identify relevant content. Similarly, several other content selection approaches [4], [5], [6] involve similar preprocessing steps. Due to these various preprocessing steps, these models cannot effectively address a change in domain. The research presented in this paper addresses this challenge by devising a stochastic content selection technique which is independent of the domain.

Our approach is based on allocating a weight to each element in the knowledge source to determine whether it should be selected or not. The weight is calculated using the relevance of the knowledge to the extracted keywords from a question. The main contributions from this research is the publicly available Java library[1] of the proposed content selection framework. We have also made available the datasets used in the experiments which can be used to further extend or customize the proposed framework.

The remainder of this paper is structured as follows. In Section II we describe the proposed model in detail. Section III describes the results from experiments used to validate the model. Section IV evaluates the results from this paper against other similar works. And Section V concludes the paper with an outlook on future work.

II. RealText$_{cs}$ model

Fig. 1 depicts an outline of proposed model. The objective of this model is to select relevant content from knowledge source for set of keywords extracted from a question. Triples are utilized as the knowledge source. A triple is a data structure containing a subject, a predicate and an object (e.g., ⟨Steve Jobs, founder, Apple Inc.⟩). Given a knowledge source of this type, the model should then be able to select relevant triples for the keywords. Selection is based on allocating a weight for each triple in knowledge source, where the weight represents how important the triple is to the domain represented by keywords. The method first assigns weight for each term in triple and then sum it up to

[1] http://staff.elena.aut.ac.nz/Parma-Nand/projects.html

calculate the total weight for the triple. The calculation of weight is accomplished by comparing weighted frequencies between a domain corpus (built using keywords) and a general reference corpus. If a term has higher frequency in domain corpus compared to general reference corpus, then it implies that term is important for the domain. The stop words were removed for this calculation. The model also consisted of two modules for filtering duplicates and finalize content.

The following sections provide a detailed overview of the subcomponents of the method.

A. Content Selection

The content selection (see Fig. 1) was implemented using three major components; term weighting, content filtering, and content finalizing. In the term weighting step, each triple is allocated a weight based on their importance for the domain. The content filtering is responsible for eliminating duplicate triples. In the content finalizing, a threshold based selection is implemented which selects the finalized content. DBpedia triple store was utilized as the knowledge source. Further information about selection and acquiring content from triple store is described in Section II-B.

1) Term weighting: The method we employed for term weighting is log likelihood distance [7], [8], [9] calculated based on two corpora; reference and domain. In particular, what we wanted to know from this calculation was the importance of a particular term to the context/domain. For this calculation a specific domain corpus is needed for each keyword set. This raised the issue of building a domain corpus based on a given set of keywords. This was accomplished using dynamic corpus building with text snippets acquired from the web. Section II-C describes this dynamic corpus building process in detail.

The calculation of weight of a term (w_t) was carried out using (1) as shown below:

$$W_t = 2 \times \left(\left(f_t^{dom} \times \log \left(\frac{f_t^{dom}}{f\_exp_t^{dom}} \right) \right) + \left(f_t^{ref} \times \log \left(\frac{f_t^{ref}}{f\_exp_t^{ref}} \right) \right) \right) \quad (1)$$

where, f_t^{dom} and f_t^{ref} represent frequency of term (t) in domain corpus and reference corpus respectively. Expected frequency of a term (t) in domain ($f\_exp_t^{dom}$) and reference corpora ($f\_exp_t^{ref}$) were calculated as follows:

$$f\_exp_t^{dom} = s_{dom} \times \left(\frac{f_t^{dom} + f_t^{ref}}{s_{dom} + s_{ref}} \right) \quad (2)$$

$$f\_exp_t^{ref} = s_{ref} \times \left(\frac{f_t^{dom} + f_t^{ref}}{s_{dom} + s_{ref}} \right) \quad (3)$$

where, s_{dom} and s_{ref} represent total number of tokens in domain corpus and reference corpus respectively. Next, we can calculate the weight of a triple as shown below:

$$W_T = \sum_{t \in T} W_t \quad (4)$$

Table I
TRIPLE WEIGHTING EXAMPLE FOR THE TRIPLE ⟨BROOKLYN BRIDGE, TYPE, SUSPENSION⟩ PROCESSED FOR THE QUESTION "WHICH RIVER DOES THE BROOKLYN BRIDGE CROSS?"

	Subject		**Predicate**	**Object**
T	Brooklyn	Bridge	type	Suspension
f_t^{dom}	21	19	0.0	0.0
f_t^{ref}	0.0	0.0	17120	1394
$f\_exp_t^{dom}$	1.0834×10^{-4}	9.8028×10^{-5}	0.8832	0.0071
$f\_exp_t^{ref}$	20.9998	18.9999	17119.9116	1393.9928
W_t	511.3370	462.6383	0.1766	0.0143
W_T	974.1662			

where W_T is the weight of triple T. However, in (4), stop words were not part of the calculation. Although, verbalization of triples can introduce stop words (ex: maintainedBy ⇒ maintained by), such stop words are unimportant in content selection.

Table I shows an example of weights calculated for a triple for the question "Which river does the Brooklyn Bridge cross?" taken from QALD-2 dataset. After calculation of total weight for a triple, we then sorted these triples based on the weight assigned. According to this method, triples with high weights should represent essential content that must be selected.

2) Content filtering: The method we explained up to now can prioritize the domain specific content required. However, it cannot filter out triples which contain same knowledge. For example, a triple like ⟨Steve Jobs, founder, Apple Inc⟩ can be present in Dbpedia resource page for Steve Jobs and as well as in the page for Apple Inc. Practical scenarios can be more complex than this. For instance, two triples such as ⟨Steve Jobs, founder, Apple Inc⟩ and ⟨Steve Jobs, co-founder, Apple Inc⟩ are also considered to be duplicates.

We implemented filtering for such triples in two steps; firstly if subject, predicate and object were similar in content between multiple triples then we removed all others keeping only one. It is not significant which one is removed as they are exact duplicates and thus have the same weight.

Next, if two of the components in triples (e.g., [subject, predicate], [subject, object], [predicate, object]) were similar in content we calculated the WordNet semantic similarity (taxonomy based) [10] for the values of remaining component. For example, if subjects and objects were similar in content between two triples, then we calculated semantic similarity between two predicate values.

Experimentally, we found the threshold value of 0.25 as the optimum semantic similarity. Elements that had a similarity factor greater than this were considered as similar in content and thus marked as duplicates. Next, we removed all other triples keeping only the triple with the highest weight.

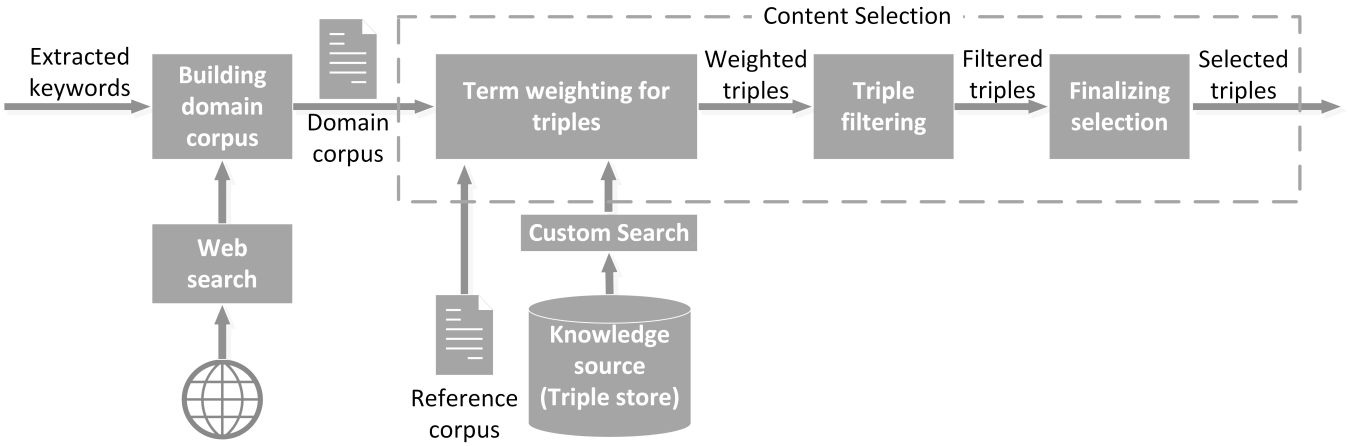

Figure 1. Schematic representation of our method

⟨Steve Jobs, birthDate, 1955-02-24⟩
⟨Steve Jobs, birthName, Steven Paul Jobs⟩
⟨Steve Jobs, board, dbpedia:Apple_Inc⟩
⟨Steve Jobs, child, dbpedia:Lisa_Brennan_Jobs⟩
⟨Steve Jobs, relative, dbpedia:Mona_Simpson⟩
⟨Steve Jobs, religion, dbpedia:Zen⟩
⟨Steve Jobs, spouse, dbpedia:Laurene_Powell_Jobs⟩
⟨Steve Jobs, nationality, American⟩
⟨Steve Jobs, alamaMater, Reed College⟩

Figure 2. Sample triples for *Steve Jobs*

3) Content finalizing: Once the content was filtered for repetitions, we than had to select the triples to represent the content. This was based on term weights that we calculated for each triple.

However, at this point we need a threshold value as the limit up to which triples should be selected for the content. We had no prior knowledge to specify such constraints, therefore, we kept this as a factor that need to be determined through experiments and detailed explanation of these experimental threshold values can be found in Section III.

B. Knowledge source and resource search

To select the content, we needed a knowledge source that could provide broad coverage of diverse areas. We employed DBpedia[2] triple store as our knowledge source. DBpedia provides knowledge about 4.0 million things, categorized under 529 classes (person, organization, places, etc.). Sample triple set for Steve Jobs is shown in Fig. 2.

Predicates in DBpedia consisted of aggregated phrases (e.g., maintainedBy, numEmployees, netIncome, etc.). We verbalized these aggregated phrases using simple rule set as these could disturb term weighting approaches. Further, in some triples, objects were mentioned as DBpedia resources (a link to another DBpedia page e.g., dbpedia:Zen). These were replaced using actual resource names (e.g., dbpedia:Zen ⇒ Zen).

In addition, we introduced another search module which could retrieve related DBpedia resource pages for specific questions. This was primarily done to reduce the search space and thus improve the overall performance. However, there is no free text search implemented for DBpedia. Query interfaces that exist for DBpedia such as SPARQL cannot accept free text queries. This was overcome by implementing a Google custom search module for Wikipedia. As DBpedia implementation is based on Wikipedia data and as both share the same resource naming convention (e.g., Wikipedia: http://www.en.wikipedia.org/wiki/Steve_Jobs and Dbpedia: http://www.dbpedia.org/resource/Steve_Jobs), we were able to acquire related DBpedia resource pages for each question using search results received from Wikipedia based Google custom web search[3] module.

C. Building domain corpus

Domain corpus is the one that changes with the keyword set being processed. For instance, when processing keywords such as "*Apple Inc., founder*", the domain corpus should consist of text related to "*Apple Inc*" and "*founder*". Selecting a keyword specific corpus of this nature is difficult to achieve. This was achieved by dynamic corpus building by retrieving snippets of texts related to the keywords.

The process of dynamic corpus building consisted of following steps. We used keywords to search web and extract

[2]http://www.dbpedia.org

[3]https://www.google.com/cse/

snippets of text. This ensured that we were getting only the content related to the question. Next, these snippets were aggregated and a domain corpus was built. This process was carried out for each question as a part of content selection.

A major issue during implementation of dynamic domain corpus building is that the domain corpus represents relatively less textual content compared to reference corpus. This raised the question of whether there is a need to perform any further enrichments of the domain corpus using general approaches such as below:

- Should terms in domain corpus be lemmatized before applying weights? e.g., should we consider frequencies for "*find*", "*finding*" and "*found*" as all share the base form of "*find*".
- Do terms in domain corpus need textual enrichment by associating them with synonyms or Wordnet Synsets [11]?

In many applications, lemmatization of verbs can increase opportunities to retrieve more information. After analysing the triples closely, we noticed that the terms that appeared in a question, more often become components of a triple as they are. However, there can be scenarios where lemmatization is important. For instance, if we consider "*who is the founder of Apple Inc.?*", the terms "*founder*" and "*found*" may typically be similar in context and thus inclusion of such terms may help our method to acquire more related knowledge.

Unlike lemmatizing, textual enrichment augments terms by associating them with synonyms or with semantically related concepts. We noticed this as an important task when searching for triples.

However, both lemmatization and Synset based enrichment were already implemented in our web search module and thus text snippets retrieved from the search module had lemmatized and enriched content. Thus, we did not attempt to further lemmatize or enrich domain corpus text.

D. Reference corpus

Balance between text genres, is an essential feature that is expected from the reference corpus. Examples of such balanced corpora are, Brown corpus, British National Corpus (BNC) among several others. The proposed approach uses BNC[4] written text subset composed of 97.3 million words, however after removing the stop words this gave us a corpus size of 52.3 million words.

Actually, what is expected from the reference corpus (BNC) is the list of tokens and their respective frequencies. Though dynamic calculation of word frequency is possible, it is computationally expensive even when the text is indexed. To avoid this our model does a preprocessing step to compute word frequencies using a unigram analysis.

[4]Other balanced corpora would have been possible, in as much as they provide a broad coverage of different categories of text.

Table II
THE RULES USED FOR PHRASE CHUNKING. PART-OF-SPEECH TAGS ARE BASED ON PENN TREEBANK GUIDELINES

Phrase chunking rule	**Definition**	**Example**
NN..	One or more adjacent noun phrase(s)	Computer
NNP..	One or more adjacent singular proper noun phrase(s)	Microsoft
NNS..	One or more adjacent plural noun phrase(s)	Undergraduates
NNPS..	One or more adjacent singular proper noun phrase(s)	Americans
[NN, NNP, NNS, NNPS][NN, NNP, NNS, NNPS]	Noun phrase combinations	Apple Incorporation
[JJ][NN, NNP, NNS, NNPS]	Adjective + One or more adjacent noun phrase(s)	Cheap computer
[JJR][NN, NNP, NNS, NNPS]	Comparative adjective + One or more adjacent noun phrase(s)	Cheaper computer
[JJS][NN, NNP, NNS, NNPS]	Superlative adjective + One or more adjacent noun phrase(s)	Cheapest computer

This preprocessing resulted in 207406 unique tokens with their respective frequencies. These tokens together with their frequencies were stored in an indexed database for efficient access.

III. EXPERIMENTAL FRAMEWORK

In the experiments we had to achieve two important goals. Firstly, we need to find a threshold value to limit the selection. Secondly, we wanted to determine whether the proposed model can retrieve relevant information from the knowledge source for the given keywords. The following sections describe the details of how these two objectives were achieved.

A. Keyword dataset

Based on our goal of the overall RealText project, we needed a keyword dataset which was extracted from a question dataset. These keywords must have a relationship with a knowledge source to perform content selection. Due to the absence of datasets of this nature, we created a keyword dataset utilizing QALD-2[5] question dataset. The creation of this keyword dataset involved a rule based noun phrase chunking technique. First, the complete question was Part-Of-Speech (POS) tagged using Stanford POS tagger [12]. The rules listed in Table II were then applied to identify key phrases mentioned in the questions.

QALD dataset is designed to evaluate QA systems that involve DBpedia as a source for extracting answers. This

[5]http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/2

fact aligns with our approach very well. For this research we selected 93 questions from initial 100 questions, eliminating 7 questions which are marked as erroneous by dataset providers.

B. Evaluation and results

To get an idea about performance of the proposed approach, we needed a general content selection method for comparison. Recently, there has been a surge in implementation of various types of content selection approaches. Among these, the statistical approach presented by Duboue and McKeown [3] is considered to be influential and has attracted the attention of many NLG researchers because of its applicability in a different range of domains. Furthermore, unlike other content selection models, Duboue and McKeown's [3] approach implemented by using semantic triples as a generalizable model which would be adapted to be used in different domains with minimal amount of preprocessing. As there are no publicly available content selection models, we implemented the model proposed by Duboue and McKeown [3] in its basic form. In essence, statistical content selection [3] works by comparing language models between source text and triple clusters. Further details on this approach can also be found in [13] and [14].

We utilized the gold standard based evaluation which is standardized by different content selection shared tasks [15], [16]. The gold triple selection is performed by selecting answers provided for QALD questions by humans and then selecting triples which are directly mentioned in these answers. These human provided answers were collected by crawling community question answer sites: *Yahoo! Answers, Answers.com, WikiAnswers,* and *AnswerBag*.

Precision (P), recall (R) and F-measure (F*) for the evaluation can be described as follows:

$$P = \frac{|triples_{selected} \cap triples_{gold}|}{|triples_{selected}|} \tag{5}$$

$$R = \frac{|triples_{selected} \cap triples_{gold}|}{|triples_{gold}|} \tag{6}$$

$$F^* = \frac{2PR}{P+R} \tag{7}$$

where, $triples_{selected}$ and $triples_{gold}$ represent triples selected by our model and triples appeared in the gold triple collection respectively.

Fig. 3 shows the experiment carried out to identify the best threshold value for selection. We measured average F-measure for 93 questions against threshold value using three sizes of the domain corpus for each question with 10 snippets, 30 snippets and 50 snippets. Table III shows statistics for the DBpedia resources and triples. Number of DBpedia resources is the total triple files selected for the question dataset. Similar triples are ones that were identified by content filtering phase as duplicates. Predetermined

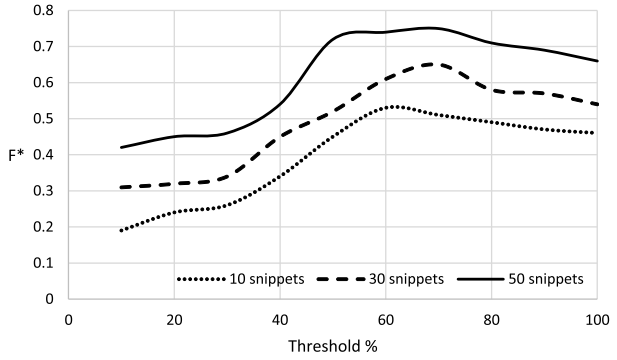

Figure 3. Average F* vs threshold for three different domain corpus sizes (in snippets)

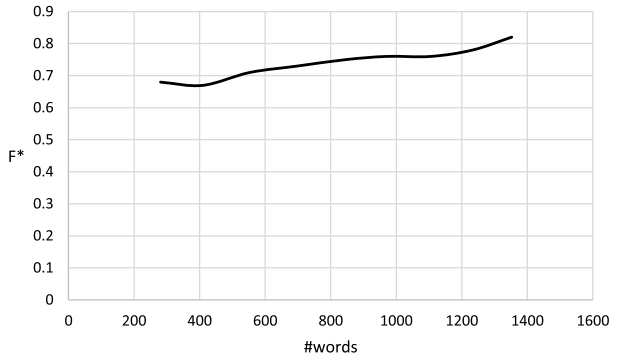

Figure 4. Average F* vs domain corpus size (in words) with threshold value 68%

Table III
STATISTICS ABOUT DBPEDIA RESOURCES AND TRIPLES PROCESSED

Number of DBpedia resources	458
Similar triples identified	78
Invalid triples (Predetermined)	1827
Invalid triples (Selected)	41

invalid triples are the triples which were initially identified as invalid (e.g., WordNet type, DBpedia Id, photo collection URL). Other invalid triples are not filtered by implemented rules and thus included in the content.

In Fig. 4, average F-measure against domain corpus size is depicted. We used threshold value of 68% for this experiment. The comparison between our approach and statistical selector [3] is given in Table IV. For this comparison, we used the experimental setting of 50 snippets (1352 words in average) as domain corpus size for each question and threshold value of 68% for selection.

C. Observations and discussions

Following observations were noticed based on results acquired in previous section.

Table IV
COMPARISON OF F* BETWEEN OUR APPROACH AND STATISTICAL SELECTOR

	Our approach	**Statistical selector**[3]
Average F*	0.74	0.56

Our initial experiment to find the best threshold value shows that threshold value ranges between 60-76% for highest F* value. We chose to use the average value of 60-76% threshold value range which is 68% as the threshold value in all subsequent evaluations. However, it should be noticed that this threshold value is dependent on accuracy of web search module and domain corpus size.

With an increase in the threshold value, there was a drop in the F* value due to decreasing precision. This is because increasing the threshold scoops in additional triples with lower weights to be included in the content.

From the graph in Fig. 4 with the increase in the domain corpus size, F* shows an upward trend. This is due to the fact that the domain corpus becomes richer in knowledge and the terms that belong to the domain start appearing more frequently thus assigning greater weight to the terms that really belong to the domain.

We also noticed that when increasing domain corpus size there could be occasions where there were slight drops in the F* value. Although, this fact is not well represented in average F*, it was noticed during the processing of the individual questions. This can be explained due to the relevancy of snippets returned by the web search module.

In our experiments with the threshold value of 68% as mentioned earlier, we noticed that domain corpus size has considerable effect on accuracy. Although the current experiments showed that the domain corpus size has positive effect on accuracy, we need to make further experiments with increasing domain corpus sizes to understand if this trend will continue and the threshold after which the effect might start showing a negative effect. When expanding the domain corpus size it will start to scope in more and more content and after eventually this expected to have a negative effect since the content will start to become more general. This can erode the significance of the domain specific terms.

Finally, based on the results shown in Table IV, it is clear that our method can perform 32% better than the statistical content selector [3]. As a comparison, Doube and McKeown's approach [3] works by creating a set of rules first whereas our approach is capable of carrying out content selection on the fly based on statistics, thus does not require the maintenance burden associated with any rule based model. However, it should be noted that the statistical selector can also retrieve general content that is included in gold standard in some scenarios. But these cases very rare compared to the complete test set.

IV. RELATED WORK

There has been a lot of work done in content selection specific to different NLG applications. Some of these approaches have already been discussed. The range of approaches that have been tried can be grouped into three different categories.

A. Machine learning and pattern recognition

Doube and McKeown's [3] approach uses language models, discussed in Section III, is a good example of the use of machine learning for content selection. The objective of their research is to acquire content selection rules that can extract content from semantic data. The rules acquired in this way are extremely domain dependent as the rule induction is performed by analysing manually retrieved domain specific content. Furthermore, rules induced are specific to the domain corpus utilized and therefore such rules cannot be applied in new domains without carrying out the some amount of pre-processing tasks. Compared to this, our approach does not require any rules to determine the content which makes it more versatile and adaptable to a wider range of domains.

Further, we considered that there exist ultimate content that need to be acquired which is represented in gold standard. However, opposite viewpoints to this idea also exist in the literature. Brazilay and Laplata [4] present a collective classification approach where they attempt to extract the content using contextual dependencies between the elements in communicative goal.

Pattern mining is also an interesting method to select the required content for non-textual data. Portet et al. [17] demonstrate the usage of pattern mining with their BT-45 system that takes neonatal intensive care data signal as the input for content selection.

B. Rule based and heuristic search

Rule based approaches are common and widely used in Content Selection because determining a rule set based on a domain can be more easily accomplished compared to designing an automatic selection process. Bouayas-Agha and Wanner [6] present a model using a predetermined rule set to acquire content. However, they also propose relevance criteria which can partially automate the process, but the usage of this criteria is not significant in the method. Though rule based systems are easy to develop, these cannot be easily adopted to different domain or cannot be used with applications that must work with different domains, for example general purpose question answering models.

C. Semantic web focused approaches

Recently, with the "web as a corpus" trend, a lot of semantic web based Content Selection models have been tried. In fact the approach presented in this paper can also be categorized in this category as it uses semantic web data

(Dbpedia) as the main source of knowledge. An example of previous similar approach is described by Doube and McKeown [3] which makes use of linked data in content selection.

The use of semantic web resources such as DBpedia and Freebase in the content selection is introduced in shared task launched in European Workshop in Natural Language Generation (ENLG) [16]. Two systems that were submitted to this shared task are based on the common ground principal model by Kutlak et al. [18] and heuristic based approach by Venigalle and Eugenio [19]. Kutlak et al. [18] hypothesise that the acquired content should contain commonly known knowledge. Based on this assumption, they propose a model which uses search engine hits for a particular knowledge element to identify whether that knowledge element is commonly known. Heuristic proposed by Venigalle and Eugenio [19] which is based on finding rules using predicate co-occurrences. However, both these aforementioned approaches did not perform well in the evaluations carried out using gold triple based evaluation.

V. Conclusions and future work

This paper presented a novel content selection framework based on a domain corpus and a general reference corpus. Central to the approach is establishing the ranking of the domain specific words that are relevant to the keywords derived from a question. This was accomplished by using a log likelihood distance calculation using the domain and the reference corpus. Based on this, each triple from our knowledge source (DBpedia triple store) is given a weight which represents the importance of the triple to the domain represented by the set of keywords. The gold standard based evaluation showed that new method can perform 32% better than cross entropy based content selection.

In the future, we plan to explore other term weighting approaches that can jointly calculate the weights in conjunction with the main log likelihood distance measure. This might be able to increase the accuracy even with a smaller domain corpus size. Furthermore, during our experiment we noticed that existing content selection models can also extract general content that sometimes become the required content, especially for questions that expect a more general content. Further, there is also a need to experiment with further increasing domain corpus sizes until the accuracy reaches a peak value. This would give us insight into the optimum domain corpus size that is suitable for a particular set of words.

References

[1] E. Reiter and R. Dale, *Building natural language generation systems*. Cambridge University Press, Jan. 2000. [Online]. Available: http://www.cambridge.org/us/academic/subjects/languages-linguistics/computational-linguistics/building-natural-language-generation-systems

[2] S. Demir, S. Carberry, and K. F. McCoy, "A discourse-aware graph-based content-selection framework," in *Sixth International Natural Language Generation Conference (INLG-10)*. Dublin, Ireland: Association for Computational Linguistics, Jul. 2010, pp. 17–25. [Online]. Available: http://dl.acm.org/citation.cfm?id=1873738.1873744

[3] P. A. Duboue and K. R. McKeown, "Statistical acquisition of content selection rules for natural language generation," in *Proceedings of the 2003 conference on Empirical methods in natural language processing -*, vol. 10. Morristown, NJ, USA: Association for Computational Linguistics, Jul. 2003, pp. 121–128. [Online]. Available: http://dl.acm.org/citation.cfm?id=1119355.1119371

[4] R. Barzilay and M. Lapata, "Collective content selection for concept-to-text generation," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*. Morristown, NJ, USA: Association for Computational Linguistics, Oct. 2005, pp. 331–338. [Online]. Available: http://dl.acm.org/citation.cfm?id=1220575.1220617

[5] C. Kelly, A. Copestake, and N. Karamanis, "Investigating content selection for language generation using machine learning," in *Twelth European Workshop on Natural Language Generation*. Athens, Greece: Association for Computational Linguistics, Mar. 2009, pp. 130–137. [Online]. Available: http://dl.acm.org/citation.cfm?id=1610195.1610218

[6] N. Bouayad-Agha, G. Casamayor, and L. Wanner, "Content selection from an ontology-based knowledge base for the generation of football summaries," in *Thirtheenth European Workshop on Natural Language Generation*. Nancy, France: Association for Computational Linguistics, Sep. 2011, pp. 72–81. [Online]. Available: http://dl.acm.org/citation.cfm?id=2187681.2187694

[7] B. F. Paul Rayson, Damon Berridge, "Extending the Cochran rule for the comparison of word frequencies between corpora," in *7th International Conference on Statistical analysis of textual data*, 2004. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.105.5529

[8] T. He, X. Zhang, and Y. Xinghuo, "An Approach to Automatically Constructing Domain Ontology," in *Pacific Asia Computational Linguistics*, Wuhan, 2006, pp. 150–157.

[9] A. Gelbukh, G. Sidorov, and L. Lavin-Villa, Eduardo Chanona-Hernandez, "Automatic Term Extraction Using Log-Likelihood Based Comparison with General Reference Corpus," in *Natural Language Processing and Information Systems*. Springer Berlin Heidelberg, 2010, pp. 248–255. [Online]. Available: http://link.springer.com/chapter/10.1007\%2F978-3-642-13881-2_26

[10] P. Pedersen, "WordNet::Similarity - Measuring the Relatedness of Concepts," in *Fifth Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, Boston, 2004, pp. 38–41.

[11] G. A. Miller, "WordNet: A Lexical Database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[12] K. Toutanova and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," in *Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics* -, vol. 13. Morristown, NJ, USA: Association for Computational Linguistics, Oct. 2000, pp. 63–70. [Online]. Available: http://dl.acm.org/citation.cfm?id=1117794.1117802

[13] P. A. Duboue and K. McKeown, "Statistical acquistion of content selection rules for ntural language generation," Columbia University, Tech. Rep., 2003.

[14] P. A. Duboue and K. R. McKeown, "ProGenIE: Biographical descriptions for intelligence analysis," in *First symposium on Intelligence and Security informatics*. Tucson: Springer-Verlag, 2003.

[15] N. Bouayad-Agha, G. Casamayor, L. Wanner, and C. Mellish, "Content selection from semantic web data," in *Seventh International Natural Language Generation Conference*. Utica IL, USA: Association for Computational Linguistics, May 2012, pp. 146–149. [Online]. Available: http://dl.acm.org/citation.cfm?id=2392712.2392745

[16] ——, "Overview of the First Content Selection Challenge from Open Semantic Web Data," in *Proceedings of the 14th European Workshop on Natural Language Generation*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 98–102. [Online]. Available: http://www.aclweb.org/anthology/W13-2112

[17] F. Portet, E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, and C. Sykes, "Automatic generation of textual summaries from neonatal intensive care data," *Artificial Intelligence*, vol. 173, no. 7-8, pp. 789–816, May 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370208002117

[18] R. Kutlak, C. Mellish, and K. van Deemter, "Content Selection Challenge - University of Aberdeen Entry," in *Fourteenth European Workshop on Natural Language Generation*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 208–209. [Online]. Available: http://www.aclweb.org/anthology/W13-2133

[19] H. Venigalla and B. D. Eugenio, "UIC-CSC: The Content Selection Challenge Entry from the University of Illinois at Chicago," in *Proceedings of the 14th European Workshop on Natural Language Generation*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 210–211. [Online]. Available: http://www.aclweb.org/anthology/W13-2134