Ontology-based Extractive Text Summarization:
The Contribution of Instances

Murillo Lagranha Flores, Elder Rizzon Santos, Ricardo Azambuja Silveira

Federal University of Santa Catarina,
Department of Informatics and Statistics, Florianópolis,
Brazil

murillo.flores@posgrad.ufsc.br, {elder.santos,ricardo.silveira}@ufsc.br

Abstract. In this paper, we present a text summarization approach focusing on multi-document, extractive and query-focused summarization that relies on an ontology-based semantic similarity measure, that specifically explores ontology instances. We employ the DBpedia Ontology and a theoretical definition of similarity to determine query-sentence and sentence-sentence similarity. Furthermore, we define an instance-linking strategy that builds the most accurate sentence representation possible while achieving a better coverage of sentences that can be represented by ontology instances. Using primarily this instances linking strategy, the semantic similarity measure and the Maximal Marginal Relevance Algorithm (MMR), we propose a summarization model that is capable of avoiding redundancy from a more fine-grained representation of sentences, due to their representation as ontology instances. We demonstrate that our summarizer is capable of achieving compelling results when compared with relevant DUC systems and recently published related studies using ROUGE metrics. Moreover, our experiments lead us to a better understanding of how ontology instances can be used to represent sentences and what is the role of said instances in this process.

Keywords. Extractive text summarization, ontologies, ontological instances.

1 Introduction

Text Summarization is the task of creating a shorter version of a document or a set of documents while keeping most of the informational content present in these documents. Automatic Text Summarizers are usually classified with regards to how they construct the final summary as either extractive or abstractive [12]. In extractive summarization, the summary is built by concatenating textual units (usually paragraphs or sentences) extracted from the original documents. Due to its conceptual simplicity and the guarantee that the sentences used in summary will be at least as legible as the sentences in the original documents extractive summarization has been a very prominent approach in automatic text summarization for the last decades.

Constructing an extractive summary that covers most of the information present in the original documents while achieving a significant reduction in length is essential to avoid redundancy. Ontology-based summarizers proposed so far explored the use of concepts as a proxy to represent the semantics of sentences, successfully avoiding redundancy and therefore achieving great results in generating extractive summaries.

However, due to their inability to distinguish between different references to the same concept, which reduces their ability to evaluate the semantics of sentences, ontology-based extractive summarizers that only explore concepts have the tendency to leave relevant sentences out of the summary for considering them redundant, when in fact they hold references to different instances of the same concept. The use of manually built ontologies makes the problem more severe, due to their reduced number of concepts.

In this paper we propose to use ontology instances to represent the semantics of sentences, attacking the problem mentioned above.

Input sentences

S1	For <u>Manchester United</u>, the reigning <u>English</u> champs, this is nothing new.
S2	<u>Manchester City</u>, on the other hand, hasn't won the <u>English</u> title since 1968.

Representation using Concepts - R1

1   [dbo:SoccerClub, dbo:Country]

2   [dbo:SoccerClub, dbo:Country]

Representation using instances - R2

1   [dbr:Manchester_United_F.C., dbr:England]

2   [dbr:Manchester_City_F.C., dbr:England]

---

dbo: dbpedia.org/ontology

dbr: dbpedia.org/resource

Fig. 1. Representation of sentences using concepts and instances defined in a ontology

Figure 1 depicts the advantages of using instances to represent the semantics of a sentence. Sentences S1 and S2 are referencing two distinct football teams from the same city, and positioning them with regards to their past performance on the English football championship. In conjunction, they are comparing and making an argument about both teams performances.

When these sentences are represented as a vector of concepts, as can be seen in R1, their representations are identical. When they are represented as a vector of instances instead, as seen in R2, their representation changes and comes closer to the real semantic differences between them.

With the goal of improving the quality of summaries built by extractive query-focused text summarizers in mind, we present an ontology-instances based summarization model. Our model uses an automatic annotation tool to link sentences to instances defined in an ontology, then uses these instances to represent the sentences and finally a semantic similarity measure to calculate the similarity between two sets of instances. We experiment on the DUC2005 dataset.

2 Representing Sentences as Concepts

To evaluate the impact that representing sentences as concepts have on the process of detecting redundant sentences, we calculated the overlap between the concepts representing distinct sentences of a summary from two summary sets. We used the DUC2004 task 2 dataset on this evaluation. In this dataset model summaries are manually created summaries, and peer summaries have been limited to include only summaries created by participating systems.

We started by employing a system for automatic annotation of DBpedia instances on text, DBpedia spotlight [5], to identify references to instances on each summary. Because the instances described in the DBpedia ontology are linked to other ontologies, we then grouped the concepts that appeared in the *rdf:type* of each instance by ontology. After that, we selected the first concept that was listed as the *rdf:type* of each group as the concept that best represented that mention on the text. With that, we created vectors of concepts that appeared in each sentence of each summary, per ontology. Those lists were considered the representation of each sentence as a vector of concepts. We consider the final result of this process to be similar to what an ontology-based summarizer that only employ concepts to represent sentences would achieve.

We calculated the intersection between the vectors of concepts representing the sentences of each summary. We classified the results in **total intersection** when the same vector of concepts represented at least two sentences of the same summary, and **partial intersection** when at least one concept appeared in two distinct vectors representing sentences of the same summary. Table 1 shows the final results. We only included results for the DBpedia and the Schema.org ontologies, as those are the larger ones linked to DBpedia instances and therefore can generate a more diverse representation of sentences.

We found that peer summaries tend to have a lower total and partial intersections than model summaries as can be seen in table 1.

Table 1. Percentage of documents with two or more sentence representations matching totally or partially on DUC 2004

Documents set	Total intersection	Partial intersection
Peer		
dbpedia	0.346	0.747
schema.org	0.343	0.705
Model		
dbpedia	0.601	0.870
schema.org	0.626	0.845

These results demonstrate that it is common to reference the same concept more than once in model summaries, created by humans. Therefore, using only concepts to detect and avoid redundancy has the potential to remove sentences that appear to be important in human-created summaries. It appears that a more granular semantic representation, that can compare the differences between sentences more precisely can achieve better results. Corroborates to this idea the fact that peer summaries have lower total and partial intersections.

3 Base Model

We use the MMR algorithm as our base model [3]. At each iteration, the algorithm selects a sentence to extract and include in the summary, until the desired length is reached. The sentence selected is always the one that is (i) more similar to the query and (ii) less redundant when compared with the previously selected sentences. We choose the MMR algorithm as our base model because it is specifically tailored for extractive query-focused summarization and can easily be extended to incorporate the advantages of a better similarity measure capable of comparing sentences and query. MMR is defined as in expression 1:

$$MMR \stackrel{def}{=} \max_{D_i \in R/S} \Bigg[\alpha(sim_1(D_i, Q))$$
$$- (1 - \alpha) \max_{D_j \in S} sim_2(D_i, D_j) \Bigg], \quad (1)$$

where Q is a query, R is a documents collection (cluster), S is the subset of documents in R already selected, R/S is the set of yet unselected documents and sim_1 and sim_2 are similarity metrics.

4 Semantic Similarity Using Instances

We extend MMR through the definition of a semantic similarity measure capable of calculating query-sentence and sentence-sentence similarity that can be used by that algorithm.

4.1 Representing Sentences as Instances

In order to determine a sentence's relevance to a specific query using ontology instances a representation of each one of them using said instances must be constructed. To this end, we employ an Instances Linking System (ILS). Instances linking systems will take snippets of text, in our case a sentence, as input and output a list of ontology instances that are mentioned in the input.

One typical problem faced by instances linking systems is the absence of detected mentions due to either a reduced number of instances defined in the underlying ontology or some inefficiency of the ILS. Instances linking systems might allow the configuration of a confidence parameter, that determines the minimum level of confidence that the ILS must have in order to link a mention, to address the problem of ILS inefficiency. Ensuring that all sentences have a valid representation is fundamental to guarantee that an instances based summarizer will be able to operate correctly, therefore we devise an approach to deal with the problem mentioned above based on the possibility of configuring a confidence parameter, as shown in algorithm 1, where $ILSLink$ is a function that links instances at a given level of confidence using the ILS.

Algorithm 1 IL with variable confidence

1: **Input** T: Text from a sentence or query.
2: **Input** c: Initial confidence value.
3: **procedure** LINK(T, c)
4:     $s \leftarrow ILSLink(T, c)$
5:     **if** $s = \emptyset$ & $c >= 0.1$ **then**
6:         $c \leftarrow c - 0.1$
7:         $s \leftarrow Link(T, c)$
8:     **return** s

4.2 Similarity Between sets of Instances

We defined sentence-sentence and query-sentence semantic similarity using their representations as ontology instances. Inspired by the work of [11] the semantic similarity between two sets of instances is defined as the average of the maximal similarity between the instances representing each one of the sets, as shown in expression 2:

$$Sim(S_1, S_2) = \frac{1}{2}\left[\frac{\sum\limits_{i_1 \in S_1} \max\limits_{i_2 \in S_2} Sim(i_1, i_2)}{|S_1|} + \frac{\sum\limits_{i_2 \in S_2} \max\limits_{i_2 \in S_1} Sim(i_1, i_1)}{|S_2|}\right], \quad (2)$$

where Sim is a semantic similarity measure between two ontology instances. We define and describe the measure used in this work in section 4.3. This definition assumes a symmetrical contribution of each one of the instance sets under comparison.

When used in conjunction with the algorithm defined in section 4.1 this definition ensures that the contribution to the overall similarity added by instances linked at a given level of confidence will not decrease with lower confidence values. In other words, its maximal similarity will not decrease with the addition of instances that are less strongly related to the sentence in question. It is worth noting, however, that the addition of said instances does increase the number of instances representing each sentence which might increase the denominator in each side of expression's 2 sum.

4.3 Similarity between Instances

We use the theoretical definition of similarity presented by [7] to define the semantic similarity measure used in this work. According to [7] "The similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to fully describe what A and B are" and can be expressed by the following equation:

$$sim(A, B) = \frac{logP(common(A, B))}{logP(description(A, B))}. \quad (3)$$

We believe that the relations that an ontology instance holds with other instances contain valuable semantic information about it. We also believe that an instance's types - the concept that they are an instance of - hold a significant amount of semantic information about it. We define that the description of an instance is formed by its types and relations with other instances, for the sake of computing its semantic similarity with other instances. Furthermore, we define that each relation will add two pieces of information to the description, separately: its type and the instance it connects to. Therefore, the description of an instance will be formed by three distinct categories of information: its **types**, its **relation types** and its **relation instances**. Figure 2 depicts two instances of different concepts that are related to each other and their description as it would be used to compute their semantic similarity.

To apply Lin's [7] definition of similarity to this work, we first define that the probability of any given component of the description of an instance is given by the probability of that component being present at a randomly selected instance of the ontology, as shown in equation 4. The probability of a relation type, for instance, is the probability of a relation of that type being present on a randomly selected instance of the ontology:

$$P(component) = \frac{count(\text{instances with component})}{count(\text{total number of instances})}. \quad (4)$$

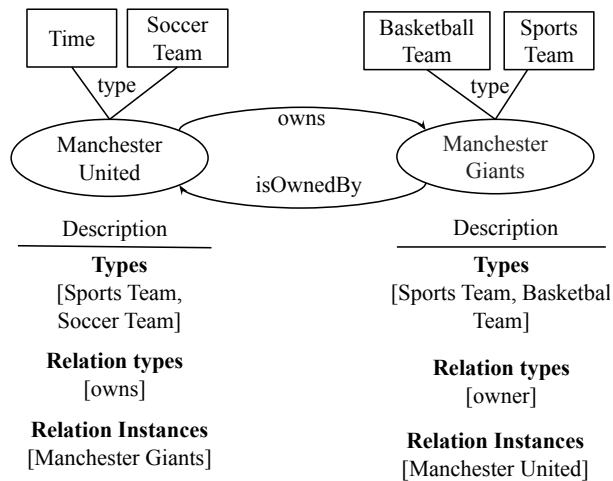We expand the definition presented in equation 4 to define the similarity of a description category

Fig. 2. Instances description example

(types, relation types and relation instances) as in equation 5, where cat is a function that returns all components of one of the categories of information in the description of the instance passed as a parameter:

$$\frac{2*\left(-1*\sum\limits_{c\in(cat(A)\cap cat(B))} logP(c)\right)}{\left(-1*\sum\limits_{c\in cat(A)} logP(c)\right)+\left(-1*\sum\limits_{c\in cat(B)} logP(c)\right)}. \quad (5)$$

Finally, the overall similarity is defined as the average of category similarities, as expressed in equation 6, where Sim_{types} is the **types** similarity, $Sim_{relTypes}$ is the **relation types** similarity and $Sim_{relInst}$ is the **relation instances** similarity:

$$sim(A,B) = \frac{1}{3}\Big[Sim_{types} + Sim_{relTypes} \\ + Sim_{relInst}\Big]. \quad (6)$$

As an example, table 2 presents the similarity between instances of the 2014 DBpedia Ontology calculated following the previous definitions.

The results in table 2 align well with the values expected from a similarity measure that follows the assumptions defined in [7]. Maximum similarity is achieved when an instance is compared against

Table 2. Similarity between 2014 DBPedia Ontology Instances

#	Measure	Value
1	sim(L.A. Lakers, L.A. Lakers)	1
2	sim(L.A. Lakers, G.S. Warriors)	0.6248
3	sim(L.A. Lakers, N.E. Patriots)	0.3958
4	sim(N.E. Patriots, S. Seahawks)	0.6301
5	sim(L.A. Lakers, Spider Man)	0.0363

itself (line 1). Teams of the same league - Los Angeles Lakers and Golden State Warriors are in the same league, as well as New England Patriots and Seattle Seahawks - have a higher similarity when compared against each other than when compared against a team in the other league (lines 2, 4 and 3). The similarities between teams in the same league have close values for both leagues (lines 2 and 4). Moreover, the similarity between a sports team and a fictional character is an order of magnitude smaller than between two sports teams on different leagues (lines 3 and 5).

5 Our Model

We extend the MMR algorithm by employing the instances linking strategy and the similarity measures defined in the previous section, as shown in figure 3.

First, instances are linked to the input documents and the query. When linking instances to the query either a fixed minimum confidence or the strategy discussed in section 4.1 are used. After that, the input documents and the query are segmented into sentences.

The MMR algorithm then uses these sentences and the instances linked to them to extract sentences and build the summary following the definition in expression 7, where S_Q are the query sentences, S_D are the documents sentences, S_S are the subset of the document sentences already selected, S_D/S_S are the yet unselected document sentences and sim is the similarity measure defined in section 4.2.
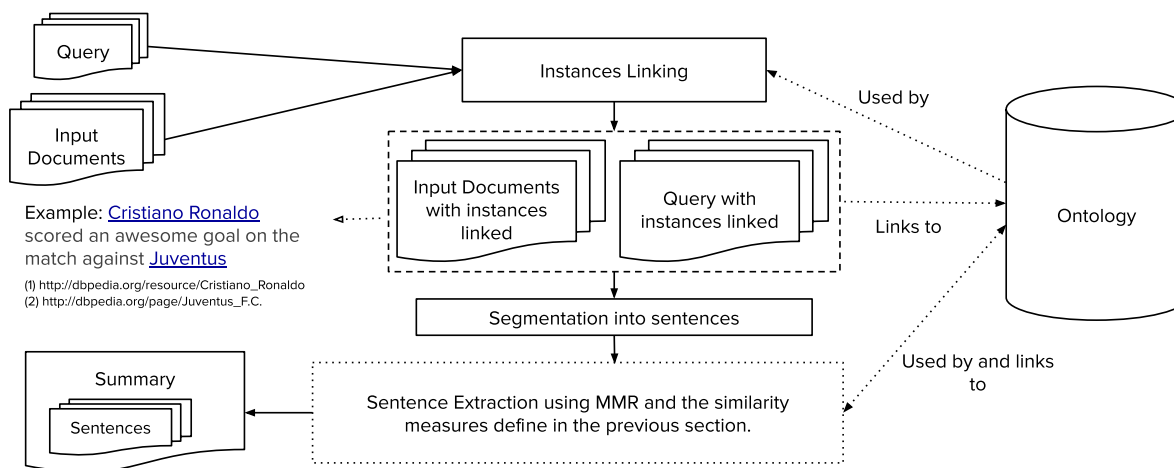
Fig. 3. The full architecture of our model

$$
MMR \stackrel{def}{=} \max_{S_i \in S_D/S_S} \left[\alpha(sim(S_i, S_Q)) - (1-\alpha) \max_{S_j \in S} sim(S_i, S_j) \right]. \quad (7)
$$

6 Related Works

Different authors have used ontologies in numerous approaches to address the extractive summarization problem.

[13] used an ontology to create a graph where each ontological concept of the document becomes a vertex and every relation between concepts becomes an edge. The most "central" sentences on that graph are extracted and establish the summary. [1] used the YAGO ontology to evaluate sentences considering a feature that expresses the sentence's popularity and pertinence, called *entityRank*. Later, sentences are extracted using a variation of MMR strategy [3]. [4] described the adopted techniques and the design of a system called Texminer, which uses ontologies in a very similar approach to the one followed by [14]. [15], heavily influenced by [8] applies an ontology to represent sentences as sets of concepts and to compute the similarity between the sentences. Closely related to our work [11] derived an approach for extractive multi-document query-focused summarization based on a semantic similarity measure that employed the WordNet Taxonomy as its knowledge-based. The authors enhanced the similarity measure with named entity semantic relatedness inferred from Wikipedia.

Different approaches that did not employ ontologies also addressed the multi-document extractive summarization problem. [2] proposed a query-focused approach based on a weighted archetypal analysis (wAA), a multivariate data representation using matrix factorization and clustering. [9] also proposed a query-focused approach suggesting to focus on three different considerations: 1) relevance, 2) coverage and 3) novelty in a probabilistic modeling framework.

Previous studies on extractive summarization have only used ontologies to capture the hierarchy of concepts in a specific domain, effectively using them as a taxonomy. Ontology instances have not been explored so far, and we are the first ones to use them to represent sentences as a way to compare these sentences semantically and enhance the summarizer's performance.

7 Experiments

In this section, we report the experiments conducted to evaluate the effectiveness of our

proposed model in multi-document query-focused extractive summarization.

7.1 Experimental Settings

We used the DUC 2005 dataset for evaluation. The DUC 2005 dataset is formed by 50 document clusters, each containing between 25 and 50 different documents on a specific topic. Each cluster has on average 31 documents and 20,236 words. The desired number of words in the summaries is 250. For each document set, between four and ten model summaries are available. This dataset was specifically created for the evaluation of multi-document query-focused summarizers.

To evaluate the summaries generated by our system quantitatively and compare them against baselines summaries as well as against summaries generated by closely related systems we use the Rouge family of metrics [6]. Rouge metrics are the *de-facto* standard in extractive summary evaluation, being widely used in the existing literature. The assessment of the quality of a summary carried out by Rouge-n metrics is based on existing model summaries (usually generated by humans) and the co-occurrence of n-grams between those model summaries and the summaries under evaluation. The evaluation follows the definition in expression 8:

$$
\begin{aligned}
&ROUGE - N \\
&= \frac{\sum\limits_{S \in \{Ref.Summ.\}} \sum\limits_{gram_n \in S} Count_{match}(gram_n)}{\sum\limits_{S \in \{Ref.Summ.\}} \sum\limits_{gram_n \in S} Count(gram_n)},
\end{aligned}
\tag{8}
$$

where N is the length of the N-gram, $Count(gram_n)$ is the number of n-grams in the reference summary and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in the summary being evaluated and a set of reference summaries (Ref. Summ.).

7.2 Implementation

To conduct the experiments we implemented our proposed model selecting the DBpedia Ontology as our base ontology. The 2014 DBpedia Ontology was built by knowledge extracted from Wikipedia and has more than four million instances defined in it [5].

We used DBpedia Spotlight to link DBpedia ontology instances to text. DBpedia Spotlight is capable of linking instances through different surface forms and with a configurable disambiguation confidence [10].

We experimented with two different variations of our system, one using a fixed value for instances linking disambiguation confidence on both documents and query and one using a variable value for the query, as described in section 4.1. We ran each variation with three initial confidence values - 0.3, 0.6, 0.9 - and three MMR α values - 0.3, 0.5, 0.7 - totalizing 18 different experimental runs.

As for computing Rouge metrics, we used the same ROUGE-1.5.5.pl Perl script used to compute the scores in the original DUC2005 competition. The parameters used were also the same ones used by DUC2005 [1].

7.3 Results

We evaluated the quality of the summaries generated by our systems using Rouge-1 and Rouge-2 as these perform better in multi-document summarization evaluation [6].

The systems name notation used in the figures describing results is defined as follows: Each system name is formed by a prefix and a suffix, separated by a dash ("-"). The prefix indicates whether that version of the system used a fixed (FIX) or a variable (VAR) confidence value when linking instances to the query. The suffix indicates the initial confidence value of the system. As an example, VAR-0.6 indicates that that version of the system ran with a variable confidence value (to link instances to the query, as described in section 4.1) starting from 0.6.

---

[1] ROUGE-1.5.5.pl -n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -d

Figure 4 shows the ROUGE-1 scores obtained by all systems, with three different values of the MMR parameter α configured. This parameter controls the balance between query relevance and summary diversity when selecting sentences. The figure shows that all systems presented better results as the instances annotation confidence decreased for values of α greater than or equal to 0.5 when query relevance had a more significant impact on sentence selection. With the α value set to 0.3 the opposite occurred - the results decreased as the confidence decreased, with a particularly acute drop between systems with confidence configured to 0.6 and 0.3. It is also worthwhile to note that except for FIX-0.9 all systems achieved their best results with α set to 0.7. These results indicate that if more relevance is given to the query, the more instances are annotated in the documents, the better. If summary diversity is given more importance when selecting sentences, more instances annotated in the documents might lead to worst results.
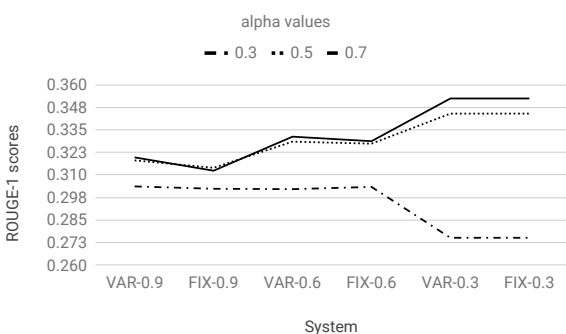

Fig. 4. Rouge-1 scores per system, with three different values of alpha

One possible explanation for achieving better performance with lower confidence and higher alpha values lives on the length difference between the query and the documents. Because the query is very short when compared to the documents, the extra instances it gets with lower values of confidence compensates the noise introduced by the extra, possibly irrelevant, instances linked to the documents. This extra instances will increase sentence extraction performance

significantly when alpha is configured to a value that gives query relevance more importance than summary diversity - or at least equals to.

Figure 5 shows that the systems with variable decreasing confidence in instances annotation on the query achieved better results than the versions with fixed confidence at the same starting level of confidence, in two occasions for a fixed α of 0.7. That corroborates with the explanation that more instances annotated on the query are more relevant to increase performance with higher values of α.
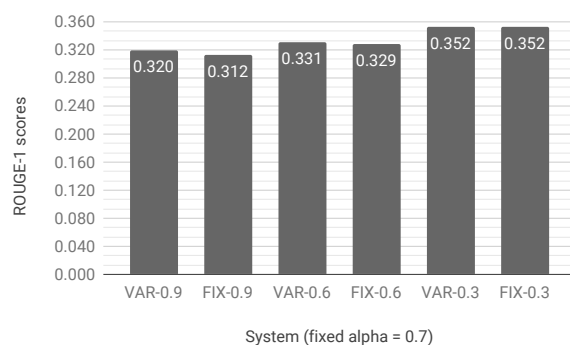
Fig. 5. Rouge-1 scores per system, with a fixed value of alpha (0.7)

Table 3 presents a comparison between the average of DUC2005 systems, closely related works and the results obtained by the best variant of our systems, VAR-0.3 system with α set to 0.7. All systems were experimented in the DUC2005 dataset. Our system outperforms the average DUC2005 systems in both compared metrics, but it also falls behind all the other systems under comparison in both metrics.

Table 3. Comparison between the average of DUC2005 systems, closely related works and our results

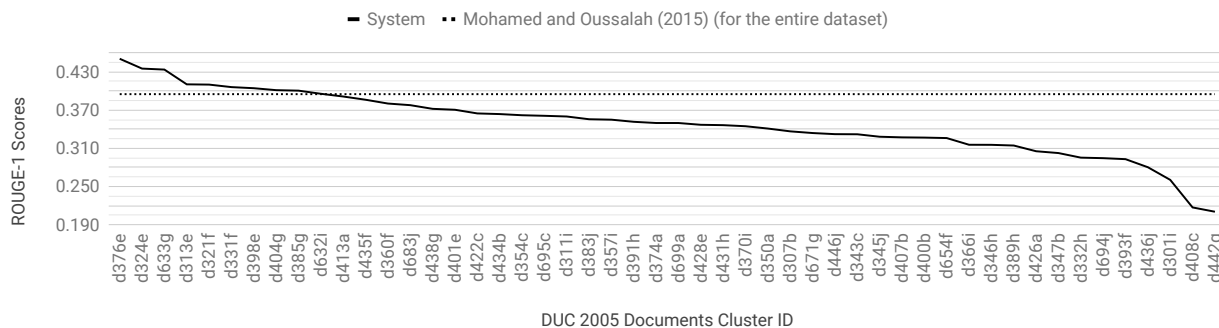| System	Rouge-1	Rouge-2
Avg. DUC2005 Systems	0.3434	0.0602
Luo et al. [9]	0.3728	0.0807
Canhasi et al. [2]	0.3945	0.0797
Mohamed et al. [11]	0.3949	0.0693
This work	0.3524	0.0639

Fig. 6. Rouge-1 scores per DUC2005 document cluster, obtained by the best variant of our system.

We also analyzed the Rouge-1 Scores obtained by the best variant of our system in all 50 DUC2005 document clusters. The results are shown in figure 6, ordered by decreasing Rouge-1 scores from left to right. To help visualize the quality of the results we also plot a line representing the best result shown in table 3 [11] for the entire dataset. As can be seen in the figure, at first the results achieved are above that line. They then decrease in a way that resembles a linear descent with a sudden drop at the end.

These results indicate that it is possible to achieve great results using instances to represent sentences and the techniques described in section 5, but further analysis is required to understand what's preventing the system from performing better on the clusters where performance is falling above the compared best.

We can draw from the conducted experiments that ontology instances can contribute to boosting the performance of extractive multi-document query-focused summarizers, by enhancing sentence-query similarity comparison and therefore helping identify sentences that are more relevant to the query. The fact that all versions of our summarizer presented better (or at least equal) results when an effort to enhance the query representation was made, by varying the instances linking confidence parameter as described in section 4.1, is an empirical evidenced of that. Following the same idea, we can also understand that the performance of summarizers based on ontology instances is highly dependent on the quantity and semantic coverage of the instances defined in the ontology and on the quality of the instances linking process. Better algorithms and similarity metrics can remediate an excess of irrelevant instances linked to the query and the documents, but they cannot remediate a lack of instances.

8 Conclusion

We proposed to use ontology instances to build an extractive query-focused multi-document summarization model, as a way to achieve a more fine-grained representation of the semantics of sentences, and avoid the problem of over-pruning sentences due to a limited semantic representation. We showed that when ontology concepts are used to represent the semantics of sentences, human-created summaries have more sentences with overlapping representations than automatically generated ones.

We extended the MMR algorithm to build our model, through an instance linking strategy with variable linking confidence and a similarity measure based on ontology instances. We experimented on the DUC2005 dataset and concluded that although representing sentences as ontology instances can help boost summarization performance further analysis is still needed to achieve better results.

References

1. **Baralis, E., Cagliero, L., Jabeen, S., Fiori, A., & Shah, S. (2013).** Multi-document summarization based on the Yago ontology. *Expert Syst. Appl.*, Vol. 40, No. 17, pp. 6976–6984.

2. **Canhasi, E. & Kononenko, I. (2014).** Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. *Expert Syst. Appl.*, Vol. 41, No. 2, pp. 535–543.

3. **Carbonell, J. & Goldstein, J. (1998).** The use of MMR, diversity-based reranking for reordering documents and producing summaries. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, ACM, New York, NY, USA, pp. 335–336.

4. **Hipola, P., Senso, J. A., Leiva-Mederos, A., & Dominguez-Velasco, S. (2014).** Ontology-based text summarization. The case of Texminer. *Library Hi Tech*.

5. **Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., & Bizer, C. (2014).** DBpedia - a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web Journal*.

6. **Lin, C.-Y. (2004).** ROUGE: A package for automatic evaluation of summaries. *Proc. ACL workshop on Text Summarization Branches Out*, pp. 10.

7. **Lin, D. (1998).** An information-theoretic definition of similarity. *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 296–304.

8. **Lin, H. & Bilmes, J. (2010).** Multi-document summarization via budgeted maximization of submodular functions. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 912–920.

9. **Luo, W., Zhuang, F., He, Q., & Shi, Z. (2013).** Exploiting relevance, coverage, and novelty for query-focused multi-document summarization. *Know.-Based Syst.*, Vol. 46, pp. 33–42.

10. **Mendes, P. N., Jakob, M., García-Silva, A., & Bizer, C. (2011).** DBpedia spotlight: Shedding light on the web of documents. *Proceedings of the 7th International Conference on Semantic Systems*, I-Semantics '11, ACM, New York, NY, USA, pp. 1–8.

11. **Mohamed, M. A. & Oussalah, M. (2015).** Similarity-based query-focused multi-document summarization using crowdsourced and manually-built lexical-semantic resources. *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 2, pp. 80–87.

12. **Nenkova, A. & McKeown, K. (2012).** *A Survey of Text Summarization Techniques*. Springer US, Boston, MA, pp. 43–76.

13. **Ramezani, M. & Feizi-Derakhshi, M.-R. (2015).** Ontology-based automatic text summarization using FarsNet. *Advances in Computer Science: an International Journal*, Vol. 4, No. 2, pp. 88–96.

14. **Umbrath, W., Wetzker, R., & Hennig, L. (2008).** An ontology-based approach to text summarization. *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, Vol. 03, No. undefined, pp. 291–294.

15. **Wu, K., Li, L., Li, J., & Li, T. (2013).** Ontology-enriched multi-document summarization in disaster management using submodular function. *Information Sciences*, Vol. 224, pp. 118 – 129.