Tackling Biomedical Text Summarization: OAQA at BioASQ 5B

Khyathi Raghavi Chandu[1] Aakanksha Naik[1] Aditya Chandrasekar[1] Zi Yang[1]
Niloy Gupta[2] Eric Nyberg[1]

Language Technologies Institute, Carnegie Mellon University
[1]{kchandu,anaik,adityac,ziy,ehn}@cs.cmu.edu
[2] niloygupta@gmail.com

Abstract

In this paper, we describe our participation in phase B of task 5b of the fifth edition of the annual BioASQ challenge, which includes answering factoid, list, yes-no and summary questions from biomedical data. We describe our techniques with an emphasis on ideal answer generation, where the goal is to produce a relevant, precise, non-redundant, query-oriented summary from multiple relevant documents. We make use of extractive summarization techniques to address this task and experiment with different biomedical ontologies and various algorithms including agglomerative clustering, Maximum Marginal Relevance (MMR) and sentence compression. We propose a novel word embedding based tf-idf similarity metric and a soft positional constraint which improve our system performance. We evaluate our techniques on test batch 4 from the fourth edition of the challenge. Our best system achieves a ROUGE-2 score of 0.6534 and ROUGE-SU4 score of 0.6536.

1 Introduction

In recent years, there has been a huge surge in the number of biomedical articles being deposited online. The National Library of Medicine (NLM) provides MEDLINE, a gigantic database of 23 million references to biomedical journal papers. Approximately 200,000 articles [1] from this database have been cited since 2015. The rapid growth of information in this centralized repository makes it difficult for medical researchers to manually find an *exact answer* for a question or to summarize the enormous content to answer a query. The problem of extracting *exact answers* for factoid questions from this data is being studied extensively, resulting in the development of several techniques including inferencing (Moldovan et al., 2002), noisy-channel transformation (Echihabi and Marcu, 2003) and exploitation of resources like WordNet (Lin and Hovy, 2003). However, recent times have also seen an interest in developing *ideal answer* generation systems which can produce relevant, precise, non-repetitive and readable summaries for biomedical questions (Tsatsaronis et al., 2015). A query based summarization system called "BioSQUASH" (Shi et al., 2007) uses domain specific ontologies like the Unified Medical Language System (UMLS) (Schuyler et al., 1993) to create a conceptual model for sentence ranking. Experiments with biomedical ontology based concept expansion and weighting techniques were conducted, where the strength of the semantic relationships between concepts was used as a similarity metric for sentence ranking (Chen and Verma, 2006). Similar methods (Yenala et al., 2015; Weissenborn et al., 2013) are used for this task where the difference lies in query similarity ranking methods.

This paper describes our efforts in creating a system that can provide ideal answers for biomedical questions. More specifically, we develop a system which can answer the kinds of biomedical questions present in the dataset for the BioASQ challenge (Tsatsaronis et al., 2015), which is a challenge on large-scale biomedical semantic indexing and question answering. We participate in Phase B of Task 5b (biomedical question-answering) for the 2016 edition of this challenge comprising of factoid, yes/no, list and summary type questions. We develop a system for biomedical summarization using MMR and clustering based techniques. To answer factoid, list and

[1] https://www.nlm.nih.gov/bsd/medline_lang_distr.html

Proceedings of the BioNLP 2017 workshop, pages 58–66,
Vancouver, Canada, August 4, 2017.

yes/no questions, we use one of the winning systems (Yang et al., 2016) from the 2015 edition of the BioASQ challenge, open-sourced after the conclusion of the challenge [2].

We build on standard techniques such as Maximal Marginal Relevance (Carbonell and Goldstein, 1998) and Sentence Compression (Filippova et al., 2015) and incorporate domain-specific knowledge using biomedical ontologies such as the UMLS metathesaurus and SNOMEDCT (Stearns et al., 2001) to build an ideal answer generator for biomedical questions. We also experiment with several similarity metrics such as jaccard similarity and a novel word embedding based tf-idf (w2v tf-idf) similarity metric within our system. We evaluate the performance of our system on the dataset for test batch 4 of the fourth edition of the challenge and report our system performance on ROUGE-2 and ROUGE-SU4 (Lin and Hovy, 2003), which are the standard metrics used for official evaluation in the BioASQ challenge. Our best system achieves ROUGE-2 and ROUGE-SU4 scores of 0.6534 and 0.6536 respectively on test batch 4 for task 4b when evaluated on *BioASQ Oracle*[3]. Various configurations and similarity metrics, granularity and algorithms selection enabled us to secure top 1,2,3 in test batch 4 and top 1,2,3,4 in test batch 5 on automatic evaluation metrics of ROUGE-2 and ROUGE-SU4, from our participation in Task 5b of ideal answer generation.

The rest of the paper is organized as follows: Section 2 describes the datasets used. In section 3, we describe our summarization pipeline, while section 4 gives a brief overview of the system used for factoid, list and yes-no questions. Section 5 presents the evaluation results of our summarization system and our observations about various system configurations. Section 6 presents a comparative qualitative error analysis of some of our system configurations. Section 7 concludes and describes future work in this area.

2 Dataset

The training data for Phase B of task 5b provides biomedical questions, where each question is associated with question type, urls of relevant PubMed articles and relevant snippets from those articles. This dataset consists of 1,799 questions. Though our ideal answer generation system is unsupervised, we use a brief manual inspection of the training data for this edition of the challenge to make an informed choice of hyperparameters for the algorithms used by our system.

To develop an ideal answer generator which can produce query-oriented summaries for each question, we can adopt one of two popular approaches: extractive or abstractive. Extractive summarization techniques choose sentences from relevant documents and combine them to form a summary. Abstractive summarization methods use relevant documents to create a semantic representation of the knowledge from these documents and then generate a summary using reasoning and natural language generation techniques. Brief analysis on a randomly sampled subset from the training data shows us that most of the sentences in the gold ideal answers are present either in the relevant snippets or relevant abstracts of PubMed articles. Hence we perform extractive summarization. We also observe an interesting ordering trend among relevant snippets which is used to develop a positional constraint. Adding this positional constraint to our similarity metrics gives us a slight boost in performance. We explain the intuition behind this idea in more detail in section 3.1.2.

For evaluation, we use the dataset from test batch 4 of the fourth edition of the BioASQ challenge which consists of 100 questions.

3 Summarization Pipeline

In this section, we describe our system pipeline for the ideal answer generation task which mainly comprises of three stages: *question-sentence relevance ranker*, *sentence selection* and *sentence tiling*. Each stage has multiple configurations depending upon various choices for algorithms, concept expansion and similarity metrics. Figure 1 shows the overall architecture of our system and also briefly mentions various algorithms used in each stage. We describe these stages and choices in more detail in subsequent sections.

3.1 Question-Sentence Relevance ranker:

In this phase, we retrieve a list of candidate sentences from gold abstracts and snippets provided for each question and compute relevance scores with respect to the question for these sentences. We can choose from several similarity metrics, biomedical ontologies and different granularities

[2] https://github.com/oaqa/bioasq

[3] http://participants-area.bioasq.org/oracle/

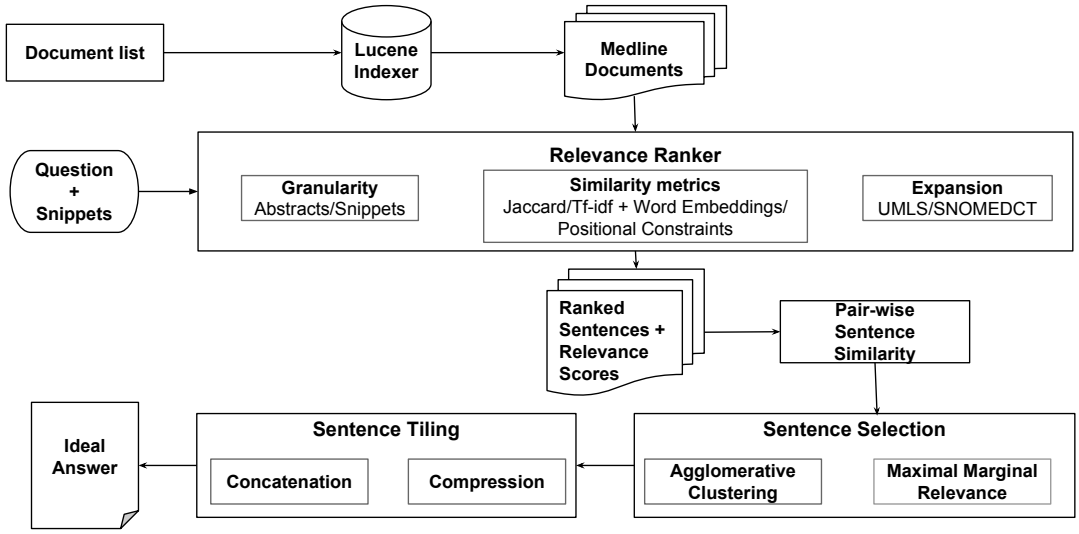

Figure 1: System pipeline for Ideal Answer Generation (with configuration choices)

for sentence scoring in this stage.

3.1.1 Granularity for Candidate Sentence Extraction

The training data provided for the BioASQ task contains a list of PubMed IDs of gold relevant documents from NLM, along with gold relevant snippets from these documents, for each question. Since, the training data only contains PubMed IDs of relevant documents, we extract complete abstract text for these documents by first indexing all Medline abstracts [4] via Lucene and then retrieving relevant documents based on PubMed IDs.

We now have two choices of granularity for candidate sentence extraction: using entire abstract texts from relevant documents or using only relevant snippets. We experiment with both possibilities. However, since relevant snippets for each question are a subset of abstract texts, which are highly relevant to the question, leveraging this insight and using only snippets for candidate sentence extraction gives us better performance, as we see from the results in Section 5.

3.1.2 Similarity metrics

The performance of both, the relevance ranker and the sentence selection phase (which is the following phase in the pipeline), depends on the similarity metrics used to capture question-sentence relevance and sentence-sentence similarity. In this section, we describe various similarity metrics which we experiment with.

Jaccard similarity: For each sentence, its relevance with respect to the question is computed as the Jaccard index between the sets containing all words occurring in the question and the sentence. This is the simplest metric which captures surface (word-level) similarity between the question and the sentence. Including related concepts obtained by concept expansion in these word sets provides some measure of semantic overlap, but this technique is not very effective as we show in section 5.

Tf-idf based similarity with word embeddings: Using ontologies such as WordNet (for general English) and UMLS/ SNOMEDCT (for biomedical domain) for concept expansion to incorporate some semantics while computing sentence similarity, is not sufficient due to the unbounded nature of such ontologies. Hence, to assimilate semantic information in a more controlled manner, we use a novel similarity metric inspired by the widely-used tf-idf cosine similarity metric which incorporates semantic information by making use of word embeddings (Mikolov et al., 2013).

Let $\mathbf{W}$ represent the symmetric word-to-word similarity matrix and $\vec{a}$, $\vec{b}$ represent tf-idf vectors for the sentences. The similarity metric is defined as:

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a}^T \mathbf{W} \vec{b}}{\sqrt{\vec{a}^T \mathbf{W} \vec{a}} \sqrt{\vec{b}^T \mathbf{W} \vec{b}}} \quad (1)$$

[4] https://www.nlm.nih.gov/databases/download/pubmed_medline.html

The word-to-word similarity matrix **W** is computed using cosine similarity between word embeddings for each word. We use word embeddings which have been pre-trained on PubMed, PMC and Wikipedia articles to incorporate domain knowledge [5].

Similarity function with positional constraints: As described in section 2, the data provided for each question contains a list of relevant abstracts of PubMed articles, as well as a list of relevant snippets extracted from these abstracts. The abstracts are ordered by relevance. Snippets on the other hand, are not ordered by relevance, but are ordered according to the abstracts that they are extracted from. Since the abstracts themselves are ordered by relevance, this gives an inherent discourse structure to the snippets. This observation motivates us to incorporate information about a snippet's position in the list into the similarity function to improve the summaries generated by our system. We first test this hypothesis using a simple baseline which gives the first snippet in the list as the summary for every question. This simple baseline is able to achieve good ROUGE scores as shown in Table 1. We experiment with two different ways of incorporating this constraint:

• **Hard positional constraint:** In this method, we enforce snippet position as a hard constraint. We achieve this by restricting the algorithm to select the first sentence of the summary from the first snippet (most relevant snippet) in the list. Remaining sentences can be selected from any snippet. This method does not have much improvement on our ROUGE scores as explained in section 5.

• **Soft positional constraint:** This method incorporates snippet position as a soft constraint by adding it to the similarity function. The augmented similarity function after incorporating snippet position is presented below:

$$positionalSim(q,s) = \alpha * sim(q,s) + (1-\alpha) * rank(s) \quad (2)$$

Here, q and s denote the question and sentence respectively; $sim(q,s)$ denotes a function which computes similarity between question and sentence (we experiment with Jaccard and tf-idf based similarities); $rank(s)$ denotes the boost given to the sentence based on the position of the snippet to which it belongs and α is a weighting parameter. The value of $rank(s)$ for a sentence is computed as follows:

$$rank(s) = 1 - pos(s)$$
$$pos(s) = snippetPos(s)/\#snippets$$

Here, $snippetPos(s)$ denotes the position (index) of the snippet, to which the sentence belongs, in the list of relevant snippets. If a sentence belongs to multiple snippets, we consider the lowest index. $\#snippets$ denotes the number of relevant snippets for the current question. This positional boost gives higher weight to sentences with lower position values (since they occur earlier in the list) and returns a normalized value in the range 0-1, to ensure that it is comparable to the range of values produced by the similarity function. Adding this constraint boosts our ROUGE scores.

3.1.3 Biomedical Tools and Ontologies

We experiment with various biomedical tools and ontologies for concept expansion, in order to incorporate relations between concepts while computing similarity. To perform concept expansion, the first step is to identify biomedical concepts from a sentence. We choose the MetaMap concept identification tool and use a python wrapper, pymetamap[6] for this purpose. This API identifies biomedical concepts from a sentence and returns a Concept Unique Identification (CUI) for each concept. This CUI acts as a unique identifier for the concept which is shared across ontologies, i.e it can be used as an ID to retrieve the same concept from the UMLS ontology. After biomedical concepts are identified, we experiment with two ontologies for concept expansion: UMLS Metathesaurus and SNOMEDCT.

- **UMLS Metathesaurus:** The UMLS Metathesarus contains many types of relations for each biomedical concept. For our task, three relation types are of interest to us: 'RB' (broader relationship), 'RL' (similar or alike relationship) and 'RQ' (related and possibly synonymous relationship). However, none of the biomedical concepts identified from questions and sentences in

[5] These pre-trained word vectors are provided by http://evexdb.org/pmresources/vec-space-models/

[6] https://github.com/AnthonyMRios/pymetamap

our training dataset contained relations of the type 'RL' or 'RQ'. Hence we perform expansion for each biomedical concept by collecting all concepts linked to it by the 'RB' relation.

- **SNOMEDCT:** The SNOMEDCT ontology does not contain CUIs for biomedical concepts. Hence, we need to use a different technique to locate concepts in this ontology. In addition to CUI, pymetamap also provides a "preferred name" for each concept. We use this preferred name to perform a full-text search in the SNOMEDCT ontology. All concepts returned by this search are then considered to be related concepts and used for expansion. Using this ontology for concept expansion returns a much larger number of related concepts, due to the nature of our search (using fuzzy text search instead of precise identifiers).

We use these techniques to perform concept expansion on both questions and sentences from relevant snippets. In Section 6, we present the results of various system configurations with and without domain specific concept expansion.

3.2 Sentence Selection

In this stage, we want to select sentences for the final summary from candidate sentences extracted by the previous stage. Since the BioASQ task has a word limit of 200, we limit the number of sentences selected for the final summary to five. This sentence limit gives us good ROUGE scores across multiple system configurations.

The simplest way of performing sentence selection is to continue selecting the sentence with the highest relevance score with respect to the question, till the sentence limit is reached. However, sentences having high relevance with respect to the question may be semantically similar, thus introducing redundancy in the generated summary. We use two algorithms to combat this issue: agglomerative clustering based on sentence similarity and Maximum Marginal Relevance (MMR) (Carbonell and Goldstein, 1998). Both algorithms require effective similarity metrics to compute semantic similarity between sentences. We experiment with various similarity metrics described in section 3.1.2. We also experiment with concept expansion using multiple biomedical ontologies.

3.2.1 Agglomerative Clustering

Redundancy reduction via clustering is one of the techniques that was proposed for biomedical query-oriented summarization (Chen and Verma, 2006). In this technique, we create all possible sentence pairs from our set of candidate sentences and compute pair-wise similarities. We then perform agglomerative clustering on the sentences using these pair-wise similarity scores. Finally, we select one sentence from each cluster to generate the final summary, in such a way that the sentence having maximum question relevance score is selected from every cluster. The number of clusters is set to the maximum number of sentences we need in the final summary (five in this case). The intuition behind this technique is that agglomerative clustering forces semantically similar sentences to fall into the same cluster. Since we only select one sentence from each cluster in the end, we discard sentences which are highly similar to the selected ones.

3.2.2 Maximal Marginal Relevance

Maximal Marginal Relevance (Carbonell and Goldstein, 1998) is a widely-used summarization algorithm which was proposed to tackle the issue of redundancy while maintaining query relevance in summarization. This algorithm selects new sentences based on a combination of relevance score with respect to the question as well as similarity score with respect to the sentences which have already been selected for the final summary. Thus, this algorithm incorporates sentence similarity as a constraint, instead of explicitly clustering sentences.

3.3 Sentence Tiling

In the final stage, we combine all selected sentences to produce the final summary. The simplest way is to append all selected sentences while constraining summary length (because of the word-limit constraint for this task). We also experiment with an LSTM-based sentence compression method. We train a neural network based on a work done previously (Filippova et al., 2015) for sentence compression. We generate training data for this network by pairing sentences from abstract texts with their full text versions. Given that this dataset is too small to train the neural network, we add in training instances from existing sentence compression data-sets. Input to this model includes the word vector representation for a word

	Experiment	ROUGE-2	ROUGE-SU4
1	Clustering + Abstract texts (with average constraint)	0.2906	0.3138
2	Clustering + Snippets (with average constraint)	0.4314	0.4347
3	Clustering + Snippets (without average constraint)	0.5609	0.5632
4	Clustering + UMLS expansion	0.5488	0.5521
5	Clustering + SNOMEDCT expansion	0.5514	0.5586
6	Clustering + UMLS expansion + weighting	0.5402	0.5431
7	Clustering + SNOMEDCT expansion + weighting	0.5530	0.5588
8	Clustering + UMLS expansion + weighted normalization	0.5592	0.5632
9	Clustering + SNOMEDCT expansion + weighted normalization	0.5585	0.5650
10	MMR	0.6338	0.6296
11	MMR + w2v tf-idf similarity	0.6168	0.6126
12	First snippet baseline	0.3363	0.3308
13	MMR + Hard positional constraint + Jaccard similarity	0.6338	0.6296
14	MMR + Soft positional constraint + Jaccard similarity	0.6419	0.6410
15	Hard positional constraint + Jaccard similarity	0.6328	0.6254
16	Soft positional constraint + Jaccard similarity	0.6433	0.6429
17	**Soft positional constraint + w2v tf-idf similarity**	**0.6534**	**0.6536**
18	MMR + tf-idf similarity + LSTM compression	0.5689	0.5723

Table 1: ROUGE scores with different algorithms, ontologies and similarity metrics

and a binary value to indicate whether the previous word was included in the output sentence. Based on these inputs, the output of the model predicts whether the word should be deleted or not. Sentences generated after word deletion are concatenated together to generate the final summary. It is to be noted that this model does not require any linguistic features.

4 Overview of system for exact answer generation

To answer factoid, list and yes/no questions, we use the publicly available system (Yang et al., 2016), which builds on participation in 2015 (Yang et al., 2015). This system uses TmTool in place of UTS (unlike (Yang et al., 2015)) for concept identification as some of the constituent parsers of TmTool identify concepts based on morphological features instead of previously coded ontologies. Also, the c-value method is used to mine frequent multi-word concepts that might not have been identified by tools such as TmTool, MetaMap and LingPipe. The idea of reranking a candidate answer based on its similarity to other candidate answers is introduced in this system for list type questions. The intuition behind this approach is that all answers to a list type question should have the same semantic type and therefore, it is useful to increase the score of a low-ranked candidate answer that has the same semantic type, and vice-versa.

Yes/No questions are answered using the technique of question inversion. The last biomedical concept present in the question is considered to be the *expected answer*. The concept mentions and tokens of the expected answer are removed from the question, which is then converted to a factoid type question. Candidate answers are generated for this factoid question using the snippets for the original question. The expected answer is then compared to the ranked list of candidate answers retrieved. The answer to the yes/no type question will be *yes* if the expected answer is among the top ranked candidate answers and *no* otherwise.

5 Evaluation and Discussion

We experiment with ideal answer generation using various system configurations which differ in similarity metrics, biomedical ontologies, sentence selection algorithms(clustering/MMR) and tiling algorithms used. The official evaluation for *ideal answers* includes manual evaluation by biomedical experts in the BioASQ team as well as automatic evaluation via ROUGE scores. To present comparable and standardized results, we run our system on the batch 4 dataset for Phase B of task 4b and get our results evaluated via the *BioASQ Oracle*. These results are shown in Table 1. We

Category	Question/Summary
Question	**What is the effect that EZH2 has on chromatin?**
Gold Ideal Answer	**Ez that catalyzes di- and trimethylation of histone H3 lysine 27 (H3K37me2/3), marks repressive to transcription.*[1]* The mammalian homologs Ezh1 and Ezh2 form similar PRC2 complexes but exhibit contrasting repressive roles.*[2]* PRC2-Ezh2 catalyzes H3K27me2/3 and its knockdown affects global H3K27me2/3 levels.*[3]*** EZH2 thus maintains chromatin in a repressive state.
Clustering + Snippets	During progenitor cell differentiation and ageing, PcG silencer EZH2 attenuates, causing loss of PRC binding and transcriptional activation of INK4b and INK4a Treg-specific ablation of Ezh2 resulted in spontaneous autoimmunity with reduced Foxp3(+) cells in non-lymphoid tissues and impaired resolution of experimental autoimmune encephalomyelitis We report that **the mammalian homologs Ezh1 and Ezh2 form similar PRC2 complexes but exhibit contrasting repressive roles.*[2]*** In this study, we found the inverse correlation between FOXP3 and Ezh2, an enzyme for histone H3K27 trimethylation (**H3K27me3**) and a central epigenetic regulator in cancer the PRC2-Ezh2 complex, which is bound to the myogenin (MyoG) promoter and muscle creatine kinase (mCK) enhancer in proliferating myoblasts, and the PRC2-Ezh1 complex, which replaces PRC2-Ezh2 on MyoG promoter in post-mitotic myotubes.
Clustering + UMLS expansion + weighted normalization	During progenitor cell differentiation and ageing, PcG silencer EZH2 attenuates, causing loss of PRC binding and transcriptional activation of INK4b and INK4a. **We report that the mammalian homologs Ezh1 and Ezh2 form similar PRC2 complexes but exhibit contrasting repressive roles**. ***[2]*** Treg-specific ablation of Ezh2 resulted in spontaneous autoimmunity with reduced Foxp3(+) cells in non-lymphoid tissues and impaired resolution of experimental **autoimmune encephalomyelitis Ez that catalyzes di- and trimethylation of histone H3 lysine 27 (H3K37me2/3) *[3]*, marks repressive to transcription.*[1]*** the PRC2-Ezh2 complex, which is bound to the myogenin (MyoG) promoter and muscle creatine kinase (mCK) enhancer in proliferating myoblasts, and the PRC2-Ezh1 complex, which replaces PRC2-Ezh2 on MyoG promoter in post-mitotic myotubes.
Clustering + SNOMEDCT expansion + weighted normalization	During progenitor cell differentiation and ageing, PcG silencer EZH2 attenuates, causing loss of PRC binding and transcriptional activation of INK4b and INK4a Treg-specific ablation of Ezh2 resulted in spontaneous autoimmunity with reduced Foxp3(+) cells in non-lymphoid tissues and impaired resolution of experimental autoimmune encephalomyelitis We report that **the mammalian homologs Ezh1 and Ezh2 form similar PRC2 complexes but exhibit contrasting repressive roles.*[2]* Ez that catalyzes di- and trimethylation of histone H3 lysine 27 (H3K37me2/3), marks repressive to transcription.*[1]*** the PRC2-Ezh2 complex, which is bound to the myogenin (MyoG) promoter and muscle creatine kinase (mCK) enhancer in proliferating myoblasts, and the PRC2-Ezh1 complex, which replaces PRC2-Ezh2 on MyoG promoter in post-mitotic myotubes.
MMR	**Ezh1 and Ezh2 maintain repressive chromatin through different mechanisms.*[1]*** The chromatin-modifying enzyme Ezh2 is critical for the maintenance of regulatory T cell identity after activation. Treg-specific ablation of Ezh2 resulted in spontaneous autoimmunity with reduced Foxp3(+) cells in non-lymphoid tissues and impaired resolution of experimental autoimmune encephalomyelitis. the PRC2-Ezh2 complex, which is bound to the myogenin (MyoG) promoter and muscle creatine kinase (mCK) enhancer in proliferating myoblasts, and the PRC2-Ezh1 complex, which replaces PRC2-Ezh2 on MyoG promoter in post-mitotic myotubes. In this study, we found the inverse correlation between FOXP3 and Ezh2, an enzyme for histone H3K27 trimethylation (H3K27me3) and a central epigenetic regulator in cancer.
MMR + w2v tf-idf	**Ezh1 and Ezh2 maintain repressive chromatin through different mechanisms.*[1]*** In this study, we found the inverse correlation between FOXP3 and Ezh2, an enzyme for histone H3K27 trimethylation (H3K27me3) and a central epigenetic regulator in cancer. These studies reveal a critical role for Ezh2 in the maintenance of Treg cell identity during cellular activation. We report that **the mammalian homologs Ezh1 and Ezh2 form similar PRC2 complexes but exhibit contrasting repressive roles.*[2]*** The chromatin-modifying enzyme Ezh2 is critical for the maintenance of regulatory T cell identity after activation.
Soft constraint + w2v tf-idf	**Ezh1 and Ezh2 maintain repressive chromatin through different mechanisms.[1]** We report that **the mammalian homologs Ezh1 and Ezh2 form similar PRC2 complexes but exhibit contrasting repressive roles.*[2]* Ez that catalyzes di- and trimethylation of histone H3 lysine 27 (H3K37me2/3), marks repressive to transcription**. During progenitor cell differentiation and ageing, PcG silencer EZH2 attenuates, causing loss of PRC binding and transcriptional activation of INK4b and INK4a. the **PRC2-Ezh2** complex, which is bound to the myogenin (MyoG) promoter and muscle creatine kinase (mCK) enhancer in proliferating myoblasts, and the PRC2-Ezh1 complex, which replaces **PRC2-Ezh2** on MyoG promoter in post-mitotic myotubes.
MMR + w2v tf-idf + LSTM sentence compression	and **ezh2 maintain repressive chromatin through different mechanisms**.[1] this study , found the inverse correlation between foxp3 and ezh2 , an enzyme for histone h3k27 trimethylation (h3k27me3) and a central epigenetic regulator in cancer . prc2-ezh2 complex , which is bound to the myogenin (myog) promoter and muscle creatine kinase (mck) enhancer in proliferating myoblasts , and the prc2-ezh1 complex , which replaces prc2-ezh2 on myog promoter in post-mitotic myotubes .

Figure 2: Summaries generated with different techniques

obtain the best results among these configurations by using soft positional constraint with tf-idf based similarity on snippets.

The first three rows in Table 1 show our experiments with different granularities for sentence extraction. While using abstract texts for sentence selection, we observe that our clustering technique frequently puts sentences with low query relevance into the same clusters. Since our selection method picks one sentence from each cluster, some sentences with low query relevance from these "bad" clusters are also selected for the final summary. To solve this issue, we imposed a constraint which filtered out sentences having a lower-than-average relevance score with respect to the question before clustering. We also tried adding this constraint while using relevant snippets, but this reduced our scores, because sentences from snippets are already relevant to the question and we end up discarding important information by filtering. We also observed that *switching granularity from abstract texts to relevant snippets significantly boosted the ROUGE scores*. Hence all subsequent experiments (rows 4-18) use snippets for sentence extraction.

Rows 4-9 show our experiments with concept expansion using various biomedical ontologies and weighting techniques. We use the following weighting technique: while calculating similarity, words from the original question and sentences carry a weight of 1, while words obtained added after concept expansion carry a weight of 0.5. *We do not observe significant gains using concept expansion*. The unbounded nature of concept expansion hurts our performance and so we refrain from using this technique in further experiments. Row 10 shows our experiment using MMR for sentence selection instead of clustering. *MMR provides a significant boost in ROUGE score*. Row 11 shows our experiment with the *w2v tf-idf based similar-*

ity metric instead of Jaccard similarity, which *decreases our ROUGE scores slightly, but is still better than previous system configurations*. Row 12 shows the scores of a baseline system which returns the first snippet from the list, which is quite high, *validating our assumption that snippet position is an important factor*. Rows 13-17 shows our experiments with different ways of adding positional constraints described in section 3.1.2. While *using a hard constraint does not show much improvement*, *soft positional constraint gives a slight boost*. Results with and without MMR for this metric are nearly comparable. *Soft constraint gives a huge boost when used with w2v tf-idf based similarity*. Row 18 shows our experiment *adding LSTM-based compression* on top of MMR with w2v tf-idf based similarity, which *reduces our scores*. Row 17 is the system configuration with the highest ROUGE score on our dataset, which uses soft positional constraint with w2v tf-idf similarity.

6 Comparative Qualitative Error Analysis

Figure 2 presents ideal answers generated by some of our system configurations for a randomly selected summary question from Task 4b Phase B data to provide a comparative qualitative error analysis. Each sentence in the ideal gold answer is indexed with a number as shown in the figure. We perform a relative analysis of the extent of information captured by a selected subset of system configurations from Table 1.

The sentence indexed [1] in the gold ideal answer is present word-for-word in summaries created by two configurations: Clustering + SNOMEDCT expansion + weighted normalization and Soft constraint + w2v tf-idf. Clustering + UMLS expansion + weighted normalization contains a longer version of this sentence. We also observe that this sentence does not contain any of the terms from the original question. Hence, summaries generated by all configurations using only Jaccard similarity (Clustering + Snippets, MMR) do not contain this sentence since there is no surface-level similarity. However, methods which incorporate some semantic information via word embeddings (w2v tf-idf similarity) or concept expansion (UMLS/ SNOMEDCT) include this sentence in the final summary, which shows that incorporating semantic information is important to bridge the vocabulary gap in some situations.

The sentence indexed [2] in the gold answer is present in summaries generated by most of the configurations as shown but with extra phrases such as 'We report that' at the beginning of the sentence. Though the presence of such words does not have a major impact on automatic scores like ROUGE, it influences the manual evaluation which also judges summary readability. However, the LSTM-based compression method removes these words via deletion. We observe that this sentence contains the concept "Ezh2" which is also present in the question. Hence, some configurations which use surface-level similarity (Clustering+Snippets) also pick this sentence for the final summary. But this sentence is not present in the summary generated by the MMR + snippets configuration. This happens because many sentences selected by the algorithm already contain the concept "Ezh2" and so this sentence is excluded due to its similarity to already selected sentences.

7 Conclusion and Future Work

In this paper, we present a system for query-oriented summary generation. Our comparison of MMR and agglomerative clustering-based techniques shows that while clustering selects distinct sentences, it is unable to select sentences with high query relevance. This can be improved by learning hyperparameters like number of clusters and number of sentences to be selected from each cluster based on the type of question. We plan to investigate this in the future. We find that unbounded concept expansion hurts our system scores. LSTM-based compression also hurts our system scores and we need to investigate upon this in the future to select the optimal parameters for compression ratio in order to maximize recall and precision. We also find that incorporating word embedding based tf-idf similarity along with soft positional constraints outperforms surface level word similarity with soft positional constraints. This is because the former captures both semantic information of the content as well as relevance to query based on sentence position.

Acknowledgments

This research was supported in parts by grants from Accenture PLC (PI: Anatole Gershman), NSF IIS 1546393 and NHLBI R01 HL122639.

References

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 335–336.

Ping Chen and Rakesh Verma. 2006. A query-based medical information summarization system using ontology knowledge. In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*. IEEE, pages 37–42.

Abdessamad Echihabi and Daniel Marcu. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 16–23.

Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *EMNLP*. pages 360–368.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pages 71–78.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Dan I Moldovan, Sanda M Harabagiu, Roxana Girju, Paul Morarescu, V Finley Lacatusu, Adrian Novischi, Adriana Badulescu, and Orest Bolohan. 2002. Lcc tools for question answering. In *TREC*.

Peri L Schuyler, William T Hole, Mark S Tuttle, and David D Sherertz. 1993. The umls metathesaurus: representing different views of biomedical concepts. *Bulletin of the Medical Library Association* 81(2):217.

Zhongmin Shi, Gabor Melli, Yang Wang, Yudong Liu, Baohua Gu, Mehdi M Kashani, Anoop Sarkar, and Fred Popowich. 2007. Question answering summarization of multiple biomedical documents. In *Advances in Artificial Intelligence*, Springer, pages 284–295.

Michael Q Stearns, Colin Price, Kent A Spackman, and Amy Y Wang. 2001. Snomed clinical terms: overview of the development process and project status. In *Proceedings of the AMIA Symposium*. American Medical Informatics Association, page 662.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics* 16(1):138.

Dirk Weissenborn, George Tsatsaronis, and Michael Schroeder. 2013. Answering factoid questions in the biomedical domain. *BioASQ@ CLEF* 1094.

Zi Yang, Niloy Gupta, Xiangyu Sun, Di Xu, Chi Zhang, and Eric Nyberg. 2015. Learning to answer biomedical factoid & list questions: Oaqa at bioasq 3b. In *CLEF (Working Notes)*.

Zi Yang, Yue Zhou, and Eric Nyberg. 2016. Learning to answer biomedical questions: Oaqa at bioasq 4b. *ACL 2016* page 23.

Harish Yenala, Avinash Kamineni, Manish Shrivastava, and Manoj Kumar Chinnakotla. 2015. Iiith at bioasq challange 2015 task 3b: Bio-medical question answering system. In *CLEF (Working Notes)*.