Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task

Gabor Melli, Yang Wang, Yudong Liu, Mehdi M. Kashani, Zhongmin Shi, Baohua Gu, Anoop Sarkar and Fred Popowich
School of Computing Science, Simon Fraser University
Burnaby, BC V5A 1S6, Canada
http://natlang.cs.sfu.ca/

Abstract

This paper describes the design of the SQUASH system, the SFU Question Answering Summary Handler, developed by members of the Natural Language Lab from the SFU School of Computing Science in order to participate in the 2005 Document Understanding Conference (DUC-2005) summarization task. The system design involves semantic role labelling, semantic subgraph-based sentence selection and automatic post-editing to create a question-based 250 word summary from a set of documents, all of which are relevant to the question topic. We also present and discuss the various evaluations performed on our system output, comparing our performance to the other systems that took part in the DUC 2005 competition.

1 Introduction

The SQUASH system, the SFU submission to the DUC 2005 Summarization track, has three main components: the Annotator module, the Synthesizer module, and the Editor module. The system starts off by annotating the documents and the question text in the **Annotator module**. These annotations are then fed to two summarization stages: The first stage which we call the **Synthesizer module**, focuses on sentence selection and sentence redundancy. The Synthesizer focuses on improving the Rouge score, while the next stage which we call the **Editor module**, focuses on linguistic readability and the human evaluation scores. In the Synthesizer module, a semantic graph is constructed based on the semantic role labelling of the documents and question text. Sentence selection is done by performing sub-graph selection on the semantic graph. Sentence redundancy is also measured and used to create sentence clusters related to the topic question. The Editor module deals with picking sentences from the sentence clusters provided by the Synthesizer and deals with issues of sentence ordering and editing out irrelevant content from long sentences. In general, the Synthesizer provides twice as many sentences compared with the 250 word length limit to each summary. The Editor is responsible for picking sentences so that appropriate sentences are picked to conform to the length limit. In addition, the Editor picks sentences based on the general vs. specific directive that is provided with each question for the topic summaries. The Editor module also performs other readability enhancements such as insertion of pronouns. This post-editing step falls halfway between summaries constructed purely using sentence selection, and full natural language generation based summary construction. Our sentence selection methods in the Synthesizer module based on semantic units in the text combined with summary post-editing phase provides a trade-off between content selection and linguistic quality in summarization. The overall system design is shown in Figure 1.

The SQUASH system is available on the web at *http://natlang.cs.sfu.ca/qa*. The current web interface to SQUASH can only be used to summarize questions on the DUC 2005 document collection (selected using the topic identifiers), to avoid running the expensive annotation step on arbitrary user-specified document collections. However, the questions themselves can be arbitrary, and not just the ones in the DUC 2005 evaluation.

2 The Annotator Module

The annotations used in our submitted system include the output of a statistical parser, a named-entity finder, and a co-reference resolver. Part-of-speech tags were extracted from the parser output. The semantic role labelling (SRL) for the documents and the question text is produced by transducing the output of a statistical parser using our own SRL system. This transduction is trained on the semantic annotations provided by the CoNLL-2005 data-set, which is a modified version of the annotation provided by the Penn PropBank data-set (Kingsbury and Palmer, 2002).

2.1 NER and Co-reference

The named-entity recognition (NER) module categorizes atomic text elements into predefined named entity classes and Co-reference Resolution (CR) provides co-reference chains between entities in the text. In our system, we used Alias-i's Lingpipe (*http://alias-i.com/lingpipe/*) system for most of the pre-processing steps (apart from semantic role labelling). Lingpipe is a package of NLP

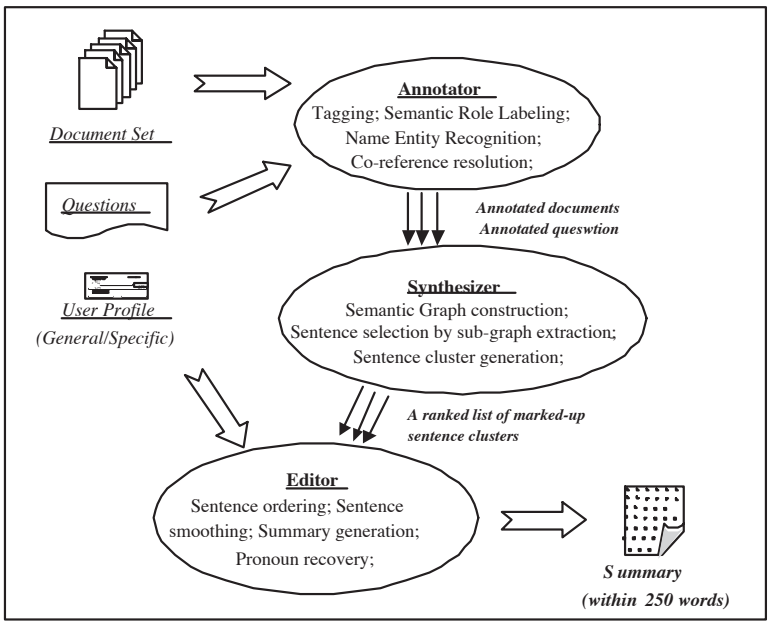

Figure 1: The overall system design of SQUASH.

tools which can be used to perform sentence boundary detection, NER, and co-reference resolution.

The named-entity classes we used are:

- Named-entity: person, organization, location
- Numeric entity: date, time, money, percent
- Pronoun entity: male, female, generic

The default named-entity (NE) model in the Lingpipe version we used, however, does not support the recognition of numeric entities. To fix this, we trained a new NE model from the MUC-7 news data corpus (MUC7, 1996) which includes annotations of named and numeric entities. Both default and new NE model were tested on a small MUC-7 news set. The precision and recall scores indicated that the new NE model does not perform as well as before on named-entities. We therefore combined results of both models for the output of the NER & CR module.

2.2 Semantic Role Labelling

A semantic role is the relationship that a syntactic constituent has with a predicate. Typical semantic roles include Agent, Patient, Instrument, etc. and also adjuncts indicating Locative, Temporal, Manner, Cause, etc. roles. We use the argument structures as defined in the Penn PropBank corpus. The task of semantic role labelling is: for each predicate in a sentence, to identify all constituents that fill a semantic role, and to determine their roles, if any (Gildea and Jurafsky, 2002; Palmer et al., 2005). Recognizing and labelling semantic arguments is a key task for answering "Who", "When", "What", "Where", "Why", and other more general types of questions in the summarization task.

Automatic semantic role labelling methods have been discussed in depth in (Gildea and Jurafsky, 2002). In the SQUASH system, the SRL component is based on an *augmented pushdown transducer* (PDT). Our semantic role labeller works with the full syntactic parses of the documents. Unlike most other SRL methods that label each constituent individually, this method finds the global best assignment of the entire sequence of SRL labels for each predicate.

There are 3 components to the SRL module: the parsing component that parses the trees produced by a statistical parser and detects the boundary for each target constituent. The probability model for argument labels is trained on the CoNLL 2005 data-set (*http://www.lsi.upc.edu/˜srlconll*). And finally the pruning component, which is implemented as a *priority queue* is used to keep track of the top n candidates for the best semantic role label sequence. We pick the top element in the priority queue as the final output. We illustrate the SRL module using an example:

Example 1
Input: (S (NP (NNP Gen) (NNP Noriega)) (VP (VBD entered) (NP (DT a) (NN conspiracy))) (. .))

Step 1: Predicate Identification
The identifier works by taking advantage of part-of-

speech tags and chunking information of the predicate.
Output: (S (NP (NNP Gen) (NNP Noriega)) (VP (VBD {*PREDICATE*} entered) (NP (DT a) (NN conspiracy))) (. .))

Step 2: Feature Extraction This operation is based on the full parse tree of the sentence. The features we use are: parse path, voice, phrase type, predicate, part-of-speech of predicate, part-of-speech of the leftmost daughter.

Step 3: Argument Label Identification Assume the target constituent is NP where (NP (DT a) (NN conspiracy)) The elements (sequences) and their probabilities in the priority queue are [1]:
(S ($_{A0}$ NP (NNP Gen) (NNP Noriega)) (VP (VBD entered) 0.7
(S ($_{A1}$ NP (NNP Gen) (NNP Noriega)) (VP (VBD entered) 0.3

By using the features of the constituent NP, the probability computation component comes up with 2 candidate semantic role labels for it based on the trained model: one is $A1$ with the probability 0.6, the other is $A2$ with the probability 0.4. Then these labels and their corresponding probabilities are combined into the current sequences in the priority queue and form the new sequences with the new probabilities.

(S ($_{A0}$ NP(NNP Gen) (NNP Noriega)) (VP (VBD entered)($_{A1}$ NP (DT a) (NN conspiracy))) 0.42
(S ($_{A0}$ NP(NNP Gen) (NNP Noriega)) (VP (VBD entered)($_{A2}$ NP (DT a) (NN conspiracy))) 0.28
(S ($_{A1}$ NP(NNP Gen) (NNP Noriega)) (VP (VBD entered)($_{A1}$ NP (DT a) (NN conspiracy))) 0.18
(S ($_{A1}$ NP(NNP Gen) (NNP Noriega)) (VP (VBD entered)($_{A2}$ NP (DT a) (NN conspiracy))) 0.12
Let $n = 2$ in this example, i.e. we keep only the top 2 elements in the priority queue. After pruning, the top elements in priority queue are:
(S ($_{A0}$ NP(NNP Gen) (NNP Noriega)) (VP (VBD entered) ($_{A1}$ NP (DT a) (NN conspiracy))) 0.42
(S ($_{A0}$ NP(NNP Gen) (NNP Noriega)) (VP (VBD entered) ($_{A2}$ NP (DT a) (NN conspiracy))) 0.28

Step 4 Final Output:
(S ($_{A0}$ NP(NNP Gen) (NNP Noriega)) (VP (VBD entered) ($_{A1}$ NP (DT a) (NN conspiracy))) (. .)) 0.42

The semantic role labeller gives the semantic relations for each sentence in each of the documents in each topic.

[1]We use Role Set defined in the *PropBank Frames scheme* (Palmer et al., 2005). Here because the constituent NP is a sister node of the predicate, all its children nodes will be labeled as NULL. In the following sequences, all the constituents that have no annotated semantic roles are labelled with the implicit label: NULL.

3 The Synthesizer Module

The task of the synthesizer module is to produce a small set of sentences from the given documents that address the provided question. The synthesizer module attempts to optimize the Rouge score. It produces twice as many sentences as necessary for the 250 word length limit on summaries to allow the subsequent Editor phase (see Section 4) to create a more human readable summary.

To accomplish this, the synthesizer uses information from the syntactic, semantic role labelling, and named entity recognition annotation. The module identifies all of the entities in the documents and assigns a value to each of them. Next, the sentences are ranked on a significance metric that is based on whether the sentence made concise use of high-valued entities found in the source documents in the proper semantic roles . Finally, sentences are iteratively selected based on their ranking. Each selection however also reduced the score (penalizes) sentences that were similar to the one selected. The synthesizer module performs the following tasks:

3.1 Semantic Graph Creation

As the first component of the SQUASH system, the primary task of the Annotator is to extract the key concepts and their relations based on the deep syntactic analyzing and semantic labelling. More specifically, the semantic relations of a document can be given by the semantic labeller. As in (Mani and Bloedorn, 1997) (cf. references cited there) a *semantic graph* is used as a visualized semantic representation of the document. Since the system is working on multiple documents for one question, the semantic graph represents the semantic relations for all the documents by sharing the common nodes and links. Here's an example of a small portion of the semantic graph constructed for topic *q0301i* from the DUC 2005 data-set. Each document in this example is assumed to contain one sentence.

Document 1: The charges stood on a US claim that Gen Noriega had entered a conspiracy with the Medelln cocaine cartel.

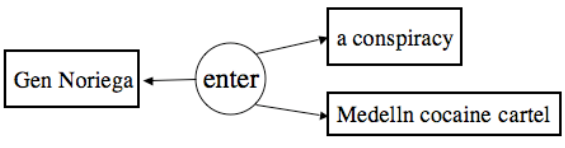

Figure 2: Semantic graph for Document 1.

Document 2: Gen Noriega had protected the cartel's operations in Panama.

Figure 3: Semantic graph for Document 2.

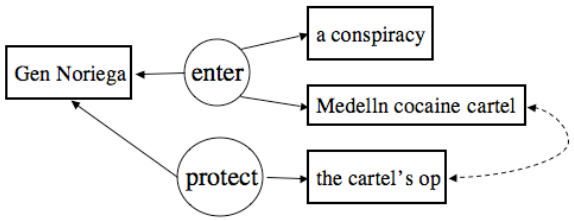

Figure 4: A merged semantic graph for multiple documents.

Constructed from the output of the Annotator, the semantic graphs contain the essential semantic relations in the document and question text. The Synthesizer module extracts the sub-structures from the graphs and performs sentence selection based on various criteria on the extracted subgraphs.

3.2 Entity Identification and Significance

A simple mechanism is used to identify the entities contained in all documents and assign to them a measure of their significance. Entity identification is based on whether a phrase was a named entity or whether a word was a noun. For example, *Gen Noriega* is marked as an entity if detected by the named-entity recognizer (NER), and *cartel* is designated an entity if the part-of-speech tagger detected that it is a noun. Similarly, the significance value to each entity is based on four factors:

- The fact that it was a named entity (0,1)
- The number of documents it was used in
- The number of sentences that it was used in.
- The number of semantic role propositions that it was used in.

Each factor is given a tunable parameter, and these parameters are first multiplied by the number of instances and then summed.

3.3 Sentence Significance Assignment

Each sentence in every document is given a significance score that will be used to rank the relevance of each sentence to the summary. The significance score is based on two metrics. The first metric assesses whether the entities in the sentence are significant. The intuition here is that between two sentences the one with proportionately more significant entities will be given preference. The second metric assesses whether the semantic roles that the entities are placed in are typical in the document set. In this case the intuition is that between two sentences the one that relates entities (related via semantic roles) in ways that are more frequently used in the document set will be given preference. The contribution based on entity significance is calculated as the summation of the individual significance values for each unique entity in the sentence. For example, if *Gen Noriega* is assigned a significance score of 0.8 and *cartel* the score 0.7 then the first score becomes 1.5. Next, this second metric is computed. Each sentence is compared against all sentences that use two or more of the same entities in the semantic graph. The significance score is increased by the amount of semantic graph overlap between the sentences. For example, if the sentence *Gen Noriega had protected the cartel's operations in Panama.* is encountered, because this matches the semantic roles of the sentence *The charges stood on a US claim that Gen Noriega had entered a conspiracy with the Medelln cocaine cartel.*, the score is increased (see Figure 4). The intuition being that a sentence that pairs these two arguments together appears to be of significance. In addition, a penalty is given for sentences with too many entities. This penalty is based on a multiplier that decreases linearly from one to zero as the number of entities increases from five to ten within the sentence. Sentences with ten or more entities are assigned a score of zero.

3.4 Avoiding Redundancy

A fixed set of prototypical sentences is sequentially selected in this step. The quantity of selected sentences was fixed to 20. This number is large enough to result in more sentences than strictly needed for the 250 word summary limit. The selection of the first sentence is simply based on largest significance score, with ties broken randomly. Next, all of the remaining sentences are compared to the selected sentence for similarity. Similar sentences are penalized in order to ensure that other interesting topics were selected. The penalty function is based on a multiplier from one to zero based on the overlap between the sentences in terms of entities and semantic role labels. If all of its entities in one sentence are included in the other along with an identical placement into semantic role arguments then the associated penalty was complete (1.0), For example, if one of the two sample sentences shown in the above example was selected, then the score for other similar sentence would be reduced to zero. This reflects the intuition that this sentence's candidacy to be assigned the role of a prototype sentence has been diminished. Each unmatched entity both in terms of presence or semantic role reduced the penalty by one half. This process is iterated until the set number of sentences was

selected. Future work will explore the use of machine learning methods as in (J. Leskovec and Milic-Frayling, 2004).

3.5 Inclusion of Similar Sentences

Finally, each of the prototypical sentences is compared anew against all of the unselected sentences. If a perfect match in entities exists then it is included in the cluster of its corresponding prototypical sentence. This step allows for the output of several similar sentences with which the Editor module could select the more appropriate version of the sentence in the summary. For example, the two sample *cartel* sentences above would be placed together in order for the Editor module to decide which version of the sentence was the most appropriate one to use.

4 The Editor Module

The task of the Editor module is to produce a summary with high linguistic quality. To achieve this, the Editor assigns a score for every sentence produced by the synthesizer module, orders all the sentences based on their scores and selects the highest scoring subset of the sentences as the candidate sentences for the summary. It also edits out irrelevant content from long sentences to generate a good quality summary with the length limit of 250 words. A series of linguistic features are used to help with ordering and editing the sentences. The Editor module performs the following steps:

- Pre-process each sentence: Recover the pronouns back to the original nominal expressions.
- Order the set of sentences based on a two-phase algorithm.
- Generate the summary based on the ordering and eliminate some irrelevant contents to increase readability.
- Resolve co-reference between entities in the summary.

4.1 Pre-processing

A major challenge in creating a coherent summary is reference resolution. Some extractive summarization methods (Blair-Goldensohn et al., 2004), perform the coreference recovery after the individual sentences are selected and put together. But since the sentences are coming from different articles, it is very hard to identify the referents of pronouns. Our method makes use of the coreference information provided by the Annotator module to first expand the pronouns into their corresponding nominal expressions in the original documents. Then in the pronoun generation step (see Section 4.5), we convert these repeated named entities back to pronouns to increase readability of the summary.

4.2 Sentence Ordering

We propose a two-phase ordering algorithm to assign the score to each sentence and order them. In the first phase, the score of each sentence is computed as a linear combination of a list of features:

$$\text{Score} = w_1 F_1 + \ldots + w_n F_n \quad (1)$$

F_i is the score of ith feature, normalized for each feature so that $F_i \in [0, 1]$. w_i is its corresponding interpolation weight. We choose these weights based on a manual study of existing summaries (we plan to learn these weights from appropriately selected training data in the future). The following is the list of features used:

- Information importance: This score is assigned by the Synthesizer module. In the Editor module, the more important the sentence is, the earlier it tends to appear in the summary.
- Location heuristics: In news wire articles, the first sentence and last sentence are often very informative and are likely to be chosen as the summary sentences for the article. We assign such sentences higher scores to increase the chance for them to appear in the earlier part of the summary.
- Sentence-length cutoff: Sentences that are too long or too short are usually not included in the summary. Here we set 30 words as the crucial point based on the statistics of average sentence length in the summary. Score for a sentence decreases linearly as its length deviates from 30 words.
- Question and sentence overlap: One of the main goals of our summary is to answer a set of questions. We assume that each sentence included in the summary should contribute to an answer of at least one question. If there are multiple questions to be answered, each of them is assigned a weight, with the first question getting the highest weight, and so on. Each sentence is assigned a score depending on possible overlap with each question, so each sentence gets several scores: one per question. Finally, a linear combination of these scores provides the ordering of sentences in the summary based on the ordering of the questions. By assigning different weight to different questions, we give a structure to the summary. The sentence ordering conforms to the order of the questions.

The DUC 2005 task requires the generated summary to be able to satisfy the granularity requirement (generic or specific) from the user profile. We include two extra features to tackle this problem:

- Named Entities: If a sentence mentions many named entities, such as time, place and people, it is more likely to contribute to specific information. So we give such sentence higher weight in the specific case and lower weight in the general case.

- Headline: Headline normally is a summary of the article, it tends to be a relatively general sentence/phrase. So here we calculate the score of information overlap between sentence and its article headline. We give higher weight to the score in the general case and lower weight in the specific case.

In the first phase, we obtain a score for each sentence and thus an ordering of all the sentences. However, this is not sufficient: to increase the coherence between two neighbouring sentences in the summary, we do a second phase of ordering in which we calculate the similarity score of two sentences based on their longest common sub-sequences. The final ordering is decided by the linear sum of the two scores. This new score provides the final ranked list of sentences.

4.3 Sentence Selection and Summary Generation

Sentences are added to the summary incrementally according to their score (computed in Section 4.2). During this step, we also delete certain words and phrases in order to include more sentences within the length limit and increase readability. Features considered are:

- Transitional words: Leading adverbial phrases and certain transitional phrases like *Interestingly, Firstly, But, And, Yet, Moreover, etc.* only make sense in the context of the original document and are probably misleading in a summary. Any such leading transitional words are removed.

- Chronological phrases such as *yesterday, in this month, next week, Monday, etc.* depend on when the news article was published. Such phrases are deleted from the sentence.

- Tagging deletion. Based on the part-of-speech tags of the sentence, we delete adverbs and adjectives before the nouns to further shorten the sentence to allow inclusion of other content in the summary.

- Pronoun penalty: Sentences that contain unresolved pronouns are removed from the sentence candidate list as they have coherence problems in a summary.

- Title elimination. For example: Mr. or Ms.

The above steps are performed on each sentence, resulting in the deletion of sentences or phrases until the summary is within the required length limit.

4.4 Redundancy Elimination

Sentences within the same cluster (from the Synthesizer module) contain potentially redundant information. The average cluster size is one to two sentences since the Synthesizer module does not include many redundant sentences within each cluster. While creating the summary by picking from the ordered list of sentences, we skip over any sentences whose cluster siblings have already been picked in the summary generation.

4.5 Pronoun Generation

As mentioned above, Lingpipe can find all the referents to a specific entity, so by running Lingpipe once on the text, we would have a chain of entities, all with the same referent. Our goal is for the latter entities to be systematically replaced by pronouns referring to the former entities. The algorithm can be divided into two phases: (a) Suggesting a replacement, and (b) Confirmation.

In the first phase, an appropriate pronoun is chosen and the text is regenerated with the specific entity replaced by this pronoun. Then, the co-reference annotation from Lingpipe is incorporated to validate the replacement. In case of valid replacement, the pronoun will remain in the final text.

In order to suggest a pronoun out of the pronoun set (he, she, his, her, him, hers), we have to deal with: Gender, and Case (nominative, accusative, possessive). These grammatical constraints on which pronoun to use are distinct from other co-reference constraints.

4.5.1 Gender Recognition

This task is performed in three consecutive phases. First, the summary is checked to see if we can resolve gender using existing referring pronouns. Second, in the annotated document set, named entity information for all the original documents exists and is used to extract gender information. Third, if some entities remain unresolved (either because they are not referred by pronoun or the co-reference is not detected by Lingpipe) a database of international frequent names is used. If the gender of an entity cannot be distinguished after these three phases, its gender is marked as *unknown*.

4.5.2 Pronoun Type

In order to choose between different case markings for pronouns (nominative, accusative, possessive), information from the parser is used. The following rules are applied:

- If most of the prepositions precede the entity and the entity is not followed by 's, the replaced pronoun should be accusative (him, her). These prepositions do not include all of the words labeled as head of a PP in the parser, so a list of frequent prepositions is also used.

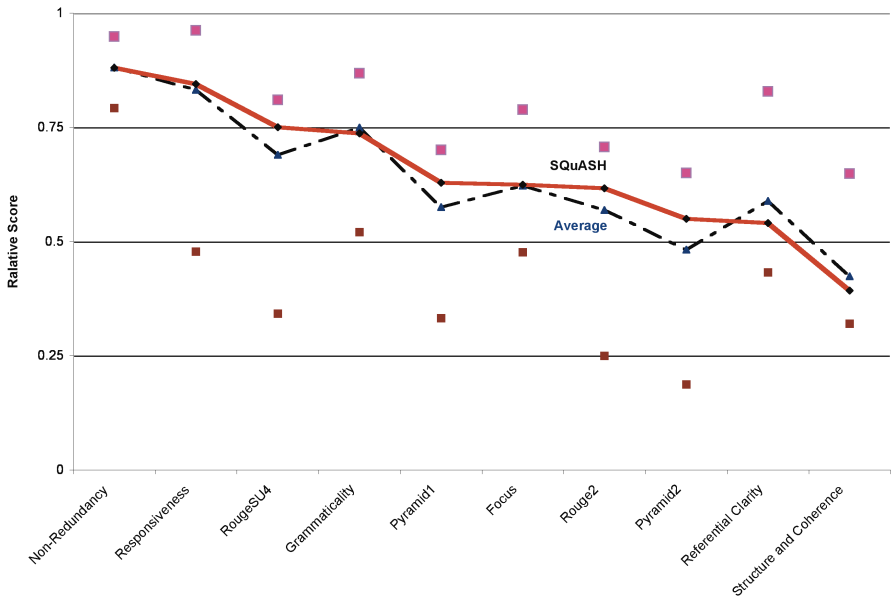

Figure 5: Our system performance compared against others: the y-axis is a normalized score relative to the best possible score for each evaluation metric, typically out of 5, but with average human scores for Rouge and the x-axis is sorted by our system score for each metric.

- If most of the prepositions precede the entity and it is followed by 's, the replaced pronoun should be possessive (his, hers).
- If a verb precedes an entity (base form, past tense, gerund, past participle, present tense) and the entity is not followed by 's, the replaced pronoun should be accusative (him, her).
- If a verb precedes an entity (base form, past tense, gerund, past participle, present tense) and the entity is followed by 's, the replaced pronoun should be possessive (his, her).
- In all other cases, the replaced pronoun is nominative and based on gender information (he, she).

4.5.3 Replacement Validation

After the pronoun is replaced in the text, the text is fed to Lingpipe. This new output is compared with the original text. If the new pronoun is still referring to the same entity that the earlier entity used to (i.e. the entity that is replaced by the pronoun), the replacement will be valid and the pronoun is kept in the text, otherwise the previous version of the text is used for the next replacement iteration. This process is repeated for all the possible combinations of co-referent entities. If the gender of entity is not known, conservatively, the process is not performed at all. This leads to lower recall in favor of higher precision. Also, if the entity is already a pronoun, nothing is done.

5 Results

We participated in both the official DUC 2005 evaluation organized by NIST and Pyramid Evaluation (PE) (Nenkova and Passonneau, 2004) on the DUC 2005 data organized by Columbia University. There are 31 systems that participated in the NIST evaluation and 25 of the systems participated in the Pyramid Evaluation. Figure 5 shows all system scores compared with the best performing system (which get a score of 1 on the y-axis). Figure 6 shows all system scores compared with the best possible score for each evaluation type: most have 5 as the best score, while the Rouge scores have the average human score as the best possible. In each case our system scores are shown by the heavy black line. In both figures we sorted the y-axis by how well we did on a particular evaluation method to explore our strengths and weaknesses. SQUASH seems to perform well above average in content selection: we have very competitive Rouge and PE scores. SQUASH ranked 7th out of the 25 systems in the F-score and 6th out of the 25 systems in the PE *modified score*. Our method of picking sentences based on scores that are derived from semantic role labels which are interpreted as *semantic graphs* seems to be well suited to the idea of evaluation based on picking semantic units that forms the core of the PE method. On some of the

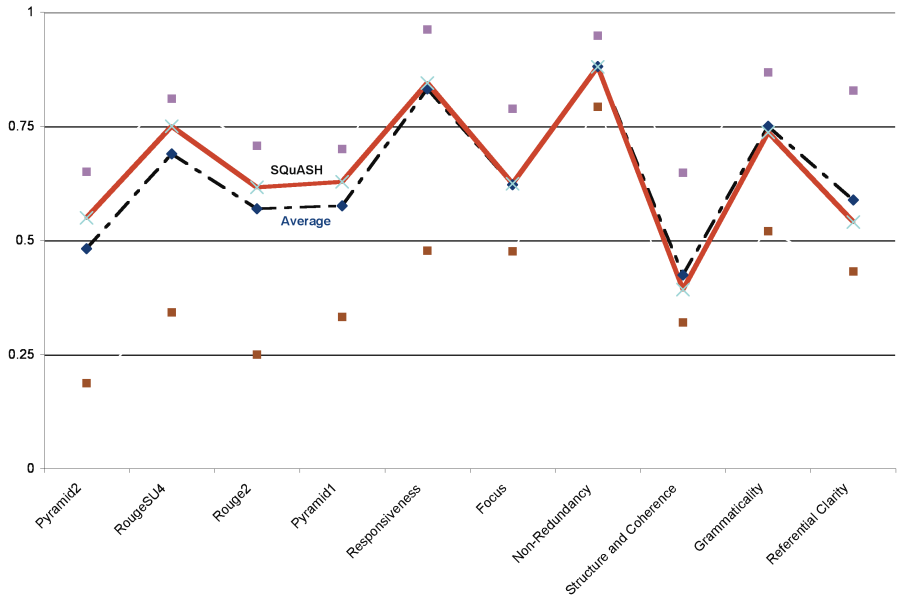

Figure 6: Our system performance compared against others: the y-axis is the same as in Fig. 5 and the x-axis is now ordered by relative rank for each metric (we consider that we did better on a metric if we placed 2nd rather than if we had a higher relative score).

linguistic quality metrics like the structure and coherence metric, our results are close to average when compared with other systems that participated in DUC 2005.

Our post-editing step falls halfway between summaries constructed purely using sentence selection, and full natural language generation based summary construction. Our sentence selection methods based on semantic units in the text combined with summary post-editing provides a trade-off between content selection and linguistic quality in summarization. We analyzed how the difference in granularity preference affects accuracy: our system does better on human readability evaluation (not counting Rouge or PE) across all topics for specific summaries as opposed to general ones.

We also built a knowledge poor baseline system (called GREEdy News Summarizer or GREENS) whose output was not submitted to the DUC 2005 evaluation. During development, we evaluated our main system against the baseline using Rouge scores on the DUC 2004 data-set. The baseline system does not emphasize readability but rather focuses on content word selection using a simple n-gram model. We do not have space here to discuss the baseline system in any detail. The table below gives a couple of the Rouge score comparisons between SQUASH and GREENS on the DUC 2005 data.

	SQUASH	GREENS
Rouge-2	0.0632	0.0435
Rouge-SU4	0.1218	0.0939

References

S. Blair-Goldensohn, D. Evans, V. Hatzivassiloglou, K. McKeown, A. Nenkova, R. Passonneau, B. Schiffman, A. Schlaikjer, A. Siddharthan, and S. Siegelman. 2004. Columbia University at DUC 2004. In *Proceedings of the Document Understanding Conference, DUC-2004*, Boston, USA.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

J. Stefan J. Leskovec and N. Milic-Frayling. 2004. Learning sub-structures of document semantic graphs for document summarization. In *LinkKDD Workshop*, pages 133–138.

P. Kingsbury and M. Palmer. 2002. From treebank to propbank. In *In Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*.

I. Mani and E. Bloedorn. 1997. Multi-document summarization by graph search and matching. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 622–628, Providence, Rhode Island.

MUC7. 1996. Message Understanding Conference (MUC) 7. LDC Catalog Id=LDC2001T02.

A. Nenkova and R. Passonneau. 2004. Evaluating content selection in summarization: the pyramid method. In *Proceedings of HLT-NAACL 2004*.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1).