Evaluation of Automatically Reformulated Questions in Question Series

Richard Shaw, Ben Solway, Robert Gaizauskas and Mark A. Greenwood
Department of Computer Science
University of Sheffield
Regent Court, 211 Portobello
Sheffield S1 4DP UK
{aca04rcs, aca04bs}@shef.ac.uk
{r.gaizauskas, m.greenwood}@dcs.shef.ac.uk

Abstract

Having gold standards allows us to evaluate new methods and approaches against a common benchmark. In this paper we describe a set of gold standard question reformulations and associated reformulation guidelines that we have created to support research into automatic interpretation of questions in TREC question series, where questions may refer anaphorically to the target of the series or to answers to previous questions. We also assess various string comparison metrics for their utility as evaluation measures of the proximity of an automated system's reformulations to the gold standard. Finally we show how we have used this approach to assess the question processing capability of our own QA system and to pinpoint areas for improvement.

1 Introduction

The development of computational systems which can answer natural language questions using large text collections as knowledge sources is widely seen as both intellectually challenging and practically useful. To stimulate research and development in this area the US National Institute of Standards and Technology (NIST) has organized a shared task evaluation as one track at the annual TExt Retrieval Conference (TREC) since 1999[1]. These evaluations began by considering factoid-type questions only (e.g. *How many calories are there in a Big Mac?*) each of which was asked in isolation to any of the others. However, in an effort to move the challenge towards a long term vision of interactive, dialogue-based question answering to support information analysts (Burger et al., 2002), the track introduced the notion of question targets and related question series in TREC2004 (Voorhees, 2005), and this approach to question presentation has remained central in each of the subsequent TRECs. In this simulated task, questions are grouped into series where each series has a target of a definition associated with it (see Figure 1). Each question in the series asks for some information about the target and there is a final "other" question which is to be interpreted as "Provide any other interesting details about the target that has not already been asked for explicitly". In this way "each series is a (limited) abstraction of an information dialogue in which the user is trying to define the target. The target and earlier questions in a series provide the context for the current question." (Voorhees, 2005).

One consequence of putting questions into series in this way is that questions may not make much sense when removed from the context their series provides. For example, the question *When was he born?* cannot be sensibly interpreted without knowledge of the antecedent of *he* provided by the context (target or prior questions). Interpreting questions in question series, therefore, becomes a critical component within a QA systems. Many QA systems have an initial document retrieval stage that takes the question and derives a query from it which is then passed to a search engine whose task is to retrieve candidate answering bearing documents for processing by the rest of the system. Clearly a question such as *When was he born?* is unlikely to retrieve documents rele-

[1] http://trec.nist.gov/

	Target 136: Shiite
Q136.1	Who was the first Imam of the Shiite sect of Islam?
Q136.2	Where is his tomb?
Q136.3	What was this persons relationship to the Prophet Mohammad?
Q136.4	Who was the third Imam of Shiite Muslims?
Q136.5	When did he die?

Figure 1: An Example Question Series

vant to answering a question about Kafka's date of birth if passed directly to a search engine. This problem can be addressed in a naive way by simply appending the target to every question. However, this has several disadvantages: (1) in some cases co-reference in a question series is to the answer of a previous question and not to the target, so blindly substituting the target is not appropriate; (2) some approaches to query formulation and to answer extraction from retrieved documents may require syntactically well-formed questions and may be able to take advantage of the extra information, such as syntactic dependencies, provided in a fully de-referenced, syntactically correct question.

Thus, it is helpful in general if systems can automatically interpret a question in context so as to resolve co-references appropriately, and indeed most TREC QA systems do this to at least a limited extent as part of their question pre-processing. Ideally one would like a system to be able to reformulate a question as a human would if they were to re-express the question so as to make it independent of the context of the preceding portion of the question series. To support the development of such systems it would useful if there were a collection of "gold standard" reformulated questions against which systems' outputs could be compared. However, to the best of our knowledge no such resource exists.

In this paper we describe the creation of such a corpus of manually reformulated questions, measures we have investigated for comparing system generated reformulations against the gold standard, and experiments we have carried out comparing our TREC system's automatic question reformulator against the gold standard and insights we have obtained therefrom.

2 The Gold Standard Corpus

Our aim was to take the questions in a TREC question series and re-express them as questions that would naturally be asked by a human asking them as a single, stand-alone question outside the context of the question series. Our intuition was that most adult native speakers would agree on a small number of variant forms these reformulated questions would take. We explored this intuition by having two persons iteratively reformulate some questions independently, compare results and evolve a small set of guidelines for the process.

2.1 Creating the Gold Standard

Ten question sets were randomly selected from sets available at `http://trec.nist.gov/data/qa/t2007_qadata.html`. These were reformulated separately by two people and results compared. From this an initial set of guidelines was drawn up. Using these guidelines another 10 question sets from the TREC 2007 QA set were independently reformulated and then the guidelines refined.

At this point the reformulators' outputs were sufficiently close to each other and the guidelines sufficiently stable that, given limited resources, it was decided reformulation could proceed singly. Using the guidelines, therefore, a further 48 question sets from 2007 were reformulated, where this time each question set was only reformulated by a single person. Each question set contained between 5 and 7 individual questions therefore around 406 questions were reformulated, creating one or more gold standard forms for each question. In total there are approximately 448 individual reformulations, with a maximum number of 3 reformulations for any single question and a mean of 1.103 reformulations per question.

2.2 Guidelines

Using the above method we derived a set of simple guidelines which anyone should be able to follow to create a set of reformulated questions.

Context independence and readability: The reformulation of questions should be understandable outside of the question series context. The reformulation should be written as a native speaker would naturally express it; this means, for example, that stop words are included.

Example: *"How many people were killed 1991* *eruption of Mount Pinatubo?"* vs *"How many people were killed in the 1991 eruption of Mount Pinatubo"*. The latter is preferred as it more readable due to the inclusion of stop words *"in the"*.

Reformulate questions so as to maximise search results:

Example: *"Who was William Shakespeare?"* vs *"Who was Shakespeare?"*. *William* should be added to the phrase as it adds extra information which could allow more results to be found.

Target matches a sub-string of the question: If the target string matches a sub-string of the question the target string should substitute the entirety of the substring. Stop-words should not be used when determining if strings and target match but should usually be substituted along with the rest of the target.

Example: Target: *"Sony Pictures Entertainment (SPE)"*; Question: *"What U.S. company did Sony purchase to form SPE?"*; Gold Standard: *"What U.S. company did Sony purchase to form Sony Pictures Entertainment (SPE)?"*

Rephrasing: A Question should not be unnecessarily rephrased.

Example: Target: *"Nissan Corp"*; Question: *"What was Nissan formerly known as?"*; *"What was Nissan Corp. formerly known as?"* is preferred over the other possible reformulation *"Nissan Corp. was formerly known as what?"*.

Previous Questions and Answers: Questions which include a reference to a previous question should be reformulated to include a PREVIOUS_ANSWER variable. Another reformulation should also be provided should a system know it needs the answer to the previous question but has not found one. This should be a reformulation of the previous question within the current question.

Example: Target: *"Harriet Miers withdraws nomination to Supreme Court"*; Question: *"What criterion did this person cite in nominating Miers?"*; Gold Standard 1: *"What criterion did PREVIOUS_ANSWER cite in nominating Harriet Miers?"*; Gold Standard 2: *"What criterion did this person who nominated Harriet Miers for the post cite in nominating Harriet Miers?"*

Targets that contain brackets: Brackets in target should be dealt with in the following way. The full target should be substituted into the question in the correct place as one of the Gold Standards. The target without the bracketed word and with it should also be included in the Gold Standard.

Example: Target: *"Church of Jesus Christ of Latter-day Saints (Mormons)"*; Question:*"Who founded the Church of Jesus Christ of Latter-day Saints?"*; Gold Standard 1: *"Who founded the Church of Jesus Christ of Latter-day Saints (Mormons)?"*; Gold Standard 2: *"Who founded the Church of Jesus Christ of Latter-day Saints?"*; Gold Standard 3 *"Who founded the Mormons?"*

Stemming and Synonyms: Words should not be stemmed and synonyms should not be used unless they are found in the target or the current question series. If they are found then both should be used in the Gold Standard.

Example: Target: *"Chunnel"*; Question:*"How long is the Chunnel?"*; Gold Standard: *"How long is the Chunnel?"*; Incorrect reformulation: *"How long is the Channel Tunnel?"*

As the term "Channel Tunnel" is not referenced in this section or hard-coded into the QA engine it cannot be substituted for "Chunnel", even though doing so may increase the probability of finding the correct answer.

It: The word *it* should be interpreted as referring to either the answer of the previous question of that set or if no answer available to the target itself.

Example:Target: *"1980 Mount St. Helens eruption"*; Question: *"How many people died when it erupted?"*; Gold Standard: *"How many people died when Mt. St. Helens' erupted in 1980?"*

Pronouns (1): If the pronouns *he* or *she* are used within a question and the TARGET is of type 'Person' then substitute the TARGET string for the pronoun. If however the PREVIOUS_ANSWER is of type 'Person' then it should be substituted instead as in this case the natural interpretation of the pronoun is to the answer of the previous question.

Example: Target: *"Jay-Z"*; Question: *"When was he born?"*; Gold Standard: *"When was Jay-Z born?"*

Pronouns (2): If the pronouns *his*/*hers*/*their* are used within a question and the TARGET is of type 'Person' then substitute the TARGET string for the pronoun appending the string "'s" to the end of the substitution. If however the PREVIOUS_ANSWER is of type 'Person' then it should be substituted as the natural interpretation of the pronoun is to the answer of the previous question.

Example: Target: *"Jasper Fforde"*; Question: *"What year was his first book written?"*; Gold Standard: *"What year was Jasper Fforde's first book written?"*

3 Evaluation against the Gold Standard

To assess how close a system's reformulation of a question in a questions series is to the gold standard requires a measure of proximity. Whatever metric we adopt should have the property that reformulations that are closer to our gold standard reformulations get a higher score. The closest possible score is achieved by getting an identical string to that of the gold standard. Following conventional practice we will adopt a metric that gives us a value between 0 and 1, where 1 is highest (i.e. a score of 1 is achieved when the pre-processed reformulation and the gold standard are identical).

Another requirement for the metric is that the ordering of the words in the reformulation is not as important as the content of the reformulation. We assume this because one key use for reformulated questions in the retrieval of candidate answer bearing documents and the presence of key content terms in a reformulation can help to find answers when it is used as a query, regardless of their order Ordering does still need to be taken into account by the metric but it should alter the score less than the content words in the reformulation.

Related to this point, is that we would like reformulations that simply append the target onto the end of the original question to score more highly on average than the original questions on their own, since this is a default strategy followed by many systems that clearly helps in many cases. These requirement can help to guide metric selection.

3.1 Choosing a metric

There are many different systems which attempt to measure string similarity. We considered a variety of tools like ROUGE (Lin, 2004) and METEOR (Lavie and Agarwal, 2007) but decided they were unsuitable for this task. ROUGE and METEOR were developed to compare larger stretches of text – they are usually used to compare paragraphs rather than sentences. We decided developing our own metric would be simpler than trying to adapt one of these existing tools.

To explore candidate similarity measures we created a program which would take as input a list of reformulations to be assessed and a list of gold standard reformulations and compare them to each other using a selection of different string comparison metrics. To find out which of these metrics best scored reformulations in the way which we expected, we created a set of test reformulations to compare against the gold standard reformulations.

Three test data sets were created: one where the reformulation was simply the original question, one where the reformulation included the target appended to the end, and one where the reformualation was identical to the gold standard. The idea here was that the without target question set should score less than the with target question set and the identical target question set should have a score of 1 (the highest possible score).

We then had to choose a set of metrics to test and chose to use metrics from the SimMetrics library as it is an open source extensible library of string similarity and distance metrics [2].

3.2 Assessing Metrics

After running the three input files against the metrics we could see that certain metrics gave a score which matched our requirements more closely than others.

Table 1 shows the metrics used and the mean scores across the data set for the different question sets. A description of each of these metrics can be found in the SimMetrics library.

From these results we can see that certain metrics are not appropriate. SmithWaterman, Jaro and JaroWinkler all do the opposite to what we require them to do in that they score a reformulation without the target higher than one with the target. This could be due to over-emphasis on word ordering. These metrics can therefore be discounted.

Levenshtein, NeedlemanWunch and QGrams-Distance can also be discounted as the difference between With target and Without target is not large enough. It would be difficult to measure improvements in the system if the difference is this small. MongeElkan can also be discounted as overall its scores are too large and for this reason it would be difficult to measure improvements using it.

Of the five remaining metrics – DiceSimilarity, JaccardSimilarity, BlockDistance, EuclideanDistance and CosineSimilarity – we decided that we should discount EuclideanDistance as it had the smallest gap between with target and without target. We now look at the other four metrics in more detail[3]:

[2] `http://www.dcs.shef.ac.uk/~sam/simmetrics.html`

[3] Refer to Manning and Schütze (2001) for more details on these algorithms.

Metric	Without Target	With Target	Identical
JaccardSim.	0.798	0.911	1.0
DiceSim.	0.872	0.948	1.0
CosineSim.	0.878	0.949	1.0
BlockDistance	0.869	0.941	1.0
EuclideanDistance	0.902	0.950	1.0
MongeElkan	0.922	0.993	1.0
Levenshtein	0.811	0.795	1.0
NeedlemanWunch	0.830	0.839	1.0
SmithWaterman	0.915	0.859	1.0
QGramsDistance	0.856	0.908	1.0
JaroWinkler	0.855	0.831	0.993
Jaro	0.644	0.589	0.984

Table 1: Mean scores across the data set for each of the different question sets.

3.2.1 Block Distance

Block Distance metric is variously named block distance, L1 distance or city block distance. It is a vector-based approach, where q and r are defined in n-dimensional vector space. The L_1 or block distance is calculated from summing the edge distances.

$$L_1(q,r) \;=\; \sum_y |\; q(y) - r(y)|$$

This can be described in two dimensions with discrete-valued vectors. When we can picture the set of points within a grid, the distance value is simply the number of edges between points that must be traversed to get from q to r within the grid. This is the same problem as getting from corner a to b in a rectilinear street map, hence the name “city-block metric”.

3.2.2 Dice Similarity

This is based on Dice coefficient which is a term based similarity measure (0-1) whereby the similarity measure is defined as twice the number of terms common to compared entities divided by the total number of terms in both. A coefficient result of 1 indicates identical vectors while a 0 indicates orthogonal vectors.

$$Dice\ Coefficient = \frac{2 * |S_1 \cap S_2|}{|S_1| + |S_2|}$$

3.2.3 Jaccard Similarity

This is a token based vector space similarity measure like the cosine distance. Jaccard Similarity uses word sets from the comparison instances to evaluate similarity. The Jaccard measure penalizes a small number of shared entries (as a portion of all non-zero entries) more than the Dice coefficient. Each instance is represented as a Jaccard vector similarity function. The Jaccard similarity between two vectors X and Y is $(X \cdot Y)/(|X||Y| - (X \cdot Y))$ where $(X \cdot Y)$ is the inner product of X and Y, and $|X| = (X \cdot X)^{1/2}$, i.e. the Euclidean norm of X. This can more easily be described as $(|X \cap Y|)/(|X \cup Y|)$

3.2.4 Cosine similarity

This is a common vector based similarity measure similar to the Dice Coefficient. The input string is transformed into vector space so that the Euclidean cosine rule can be used to determine similarity. The cosine similarity is often paired with other approaches to limit the dimensionality of the problem. For instance with simple strings a list of stopwords is used to reduce the dimensionality of the comparison. In theory this problem has as many dimensions as terms exist.

$$\cos(q,r) \;=\; \frac{\sum_y \; q(y)r(y)}{\sqrt{\sum_y q(y)^2 \sum_y r(y)^2}}$$

3.3 Using bigrams and trigrams

All four of these measures appear to value the content of the strings higher than ordering which is what we want our metric to do. However the scores are quite large, and as a result we considered refining the metrics to give scores that are not as close to 1. To do this we decided to try and increase the importance of ordering by also taking into account shared bigrams and trigrams. As we do not want ordering to be too important in our metric we introduced a weighting mechanism into the program to

Metric	Without Target	With Target	ΔGap
Dice	0.872	0.948	+0.076
Cosine	0.878	0.949	+0.071
Jaccard	0.798	0.911	+0.113
Block	0.869	0.941	+0.072

Table 2: Results for Unigram weighting

Metric	Without Target	With Target	ΔGap
Dice	0.783	0.814	-3.6
Cosine	0.789	0.816	-3.5
Jaccard	0.698	0.748	-5.5
Block	0.782	0.811	-3.5

Table 3: U:1, B:1, T:0

allow us to used a weighted combination of shared unigrams, bigrams and trigrams.

The results for just unigram weighting is shown in Table 2.

We began by testing the metrics by introducing just bigrams to give us an idea of what effect they would have. A weight ratio of U:1, B:1, T:0 was used (where U:unigram, B:bigram, T:trigram). The results are shown in Table 3.

The Δ Gap column is the increase in the difference between Without Target and With Target from the first test run which used only unigrams.

The introduction of bigrams decreases the gap between Without Target and With Target. It also lowers the scores which is good as it is then easier to distinguish between perfect reformulations and reformulations which are close but not perfect. This means that the introduction of bigrams is always going to decrease a system's ability to distinguish between Without Target and With Target. We had to now find the lowest decrease in this gap whilst still lowering the score of the with target result.

From the results of the bigrams we expected that the introduction of trigrams would further decrease the gap ($U : 1, B : 1, T : 1$). The results proved this and are shown in Table 4.

Metric	Without Target	With Target	ΔGap
Dice	0.725	0.735	-6.4
Cosine	0.730	0.735	-6.3
Jaccard	0.639	0.663	-9.0
Block	0.724	0.733	-6.1

Table 4: U:1, B:1, T:1

Metric	Without Target	With Target	ΔGap
Dice	0.754	0.770	-4.8
Cosine	0.759	0.771	-4.9
Jaccard	0.664	0.694	-7.4
Block	0.753	0.767	-4.6

Table 5: U:1, B:2

Metric	Without Target	With Target	ΔGap
Dice	0.813	0.859	-2.4
Cosine	0.819	0.860	-2.4
Jaccard	0.731	0.802	-3.7
Block	0.811	0.854	-2.2

Table 6: U:2, B:1

The introduction of trigrams has caused the gaps to significantly drop. It has also lowered the scores too much. From this evidence we decided trigrams are not appropriate to use to refine these metrics.

We now had to try and find the best weighting of unigram to bigram that would lower the With Target score from 1.0 whilst still keeping the gap between Without Target and With Target high.

We would expect that further increasing the bigram weighting would further decrease the gap and the With Target score. The results in Table 5 show this to be the case. However this has decreased the gap too much. The next step was to look at decreasing the weighting of the bigrams.

Table 6 shows that the gap has decreased slightly but the With Target score has decreased by around 10% on average. The Jaccard score for this run is particularly good as it has a good gap and is not too close to 1.0. The Without Target is also quite low which is what we want.

$U : 2, B : 1$ is currently the best weighting found with the best metric being Jaccard. Further work in this area could be directed at further modifying these weightings using machine learning techniques to refine the weightings using linear regression.

4 Our system against the Metric

Our current pre-processing system takes a question and its target and looks to replace pronouns like “he”, “she” and certain definite nominals with the target and also to replace parts of the target with the full target (Gaizauskas et al., 2005). Given our choice of metric we would hope that this strat-

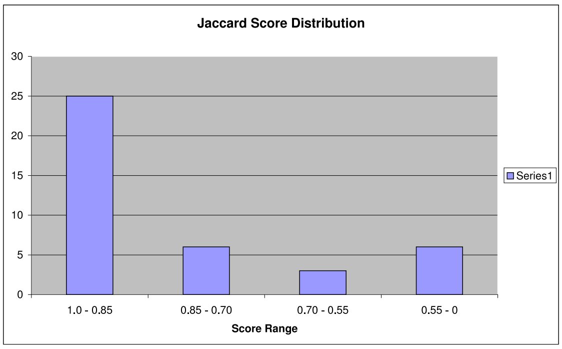

Figure 2: Graph of Jaccard score distribution

egy gets a better score than just adding the target on the end, as the ordering of the words is also taken into account by our pre-processing as it tries to achieve natural reformulations like those of our gold standard. We would therefore expect that it achieves at least the same score as adding the target on the end, which is its default strategy when no co-reference can be determined, though of course incorrect coreference resolutions will have a negative effect. One of the aims of creating the gold standard and a comparison metric was to quickly identify whether strategies such as ours are working and if not where not.

A subset of the gold standard was preprocessed by our system then compared against the results of doing no reformulation and of reformulating by simply appending the target.

Tables 7 and 8 shows how our system did in comparison. Diff shows the difference between WithTarget and Our System. Table 7 is results for weighting $U : 1, B : 0, T : 0$, Table 8 is results for $U : 2, B : 1, T : 0$.

Our system does do better than just adding the target on the end, and this difference is exaggerated (Table 8) when bigrams are taken into account, as expected since this weighting increases the metric's sensitivity to recognising our system's ability to put the target in the correct place.

Mean scores across a data set tell part of the story, but to gain more insight we need to examine the distribution of scores and then, in order to improve the system, we need to look at questions which have a low score and work out what has gone wrong. Figure 2 shows the distribution of Jaccard scores across the test data set. Looking at the scores from the data set using the U:2,B:1,T:0 weighting we find that the minimum Jaccard score was 0.44 and was for the following example:

Metric	Score
Dice	0.574
Cosine	0.578
Jaccard	0.441
Block	0.574

Table 9: Finding Bad Reformulations

Target: *"Hindenburg disaster"*; Question: *"How many of them were killed"*; Our System: *"How many of Hindenburg disaster were killed"*; Gold Standard: *"How many people were killed during the Hindenburg disaster"*.

The results of comparing our system with the gold standard for this question for all four metrics are shown in Table 9.

The problem here is that our system has wrongly replaced the term "them" with the target when in fact its antecedent was in the previous question in the series *How many people were on board?*. Once again the low score has helped us to quickly identify a problem: the system is only interpreting pronouns as references to the target, which is clearly insufficient. Furthermore should the preprocessing system be altered to address a problem like this the gold system and scoring software can be used for regression testing to ensure no previously correct reformulations have been lost.

Another example of a poor scoring reformulation is:

Target: *"Hindenburg disaster"*; Question: *"What type of craft was the Hindenburg"*; Our System: *"What type of craft was the Hindenburg disaster"*; Gold Standard: *"What type of craft was the Hindenburg"*.

For this example Jaccard gave our system reformulation a score of 0.61. The problem here is our system blindly expanded a substring of the target appearing in the question to the full target without recognizing that in this case the substring is not an abbreviated reference to the target (an event) but to an entity that figured in the event.

5 Conclusions and Future Work

In this paper we have presented a Gold Standard for question reformulation and an associated set of guidelines which can be used to reformulate other questions in a similar fashion. We then evaluated metrics which can be used to assess the effectiveness of the reformulations and validated the whole approach by showing how it could be used to help

Metric	Without Target	With Target	Our System	Diff
Dice	0.776	0.901	0.931	+3.1
Cosine	0.786	0.904	0.936	+3.1
Jaccard	0.657	0.834	0.890	+5.5
Block	0.772	0.888	0.920	+4.2

Table 7: How our system compared, U:1,B:0,T:0

Metric	Without Target	With Target	Our System	Diff
Dice	0.702	0.819	0.889	+8.7
Cosine	0.742	0.822	0.893	+9.2
Jaccard	0.616	0.738	0.839	+12.3
Block	0.732	0.812	0.884	+9.1

Table 8: How our system compared, U:2,B:1,T:0

improve the question pre-processing component of a QA system.

Further work will aim to expand the Gold Standard to at least 1000 questions, refining the guidelines as required. The eventual goal is to incorporate the approach into an evaluation tool such that a developer would have a convenient way of evaluating any question reformulation strategy against a large gold standard. Of course one also needs to develop methods for observing and measuring the effect of question reformulation within question pre-processing upon the performance of downstream components in the QA system, such as document retrieval.

References

Burger, J., C. Cardie, V. Chaudhri, R. Gaizauskas, S. Harabagiu, D. Israel, C. Jacquemin, C-Y. Lin, S. Maiorano, G. Miller, D. Moldovan, B. Ogden, J. Prager, E. Riloff, A. Singhal, R. Shrihari, T. Strzalkowski, E. Voorhees, and R. Weischedel. 2002. Issues, tasks and program structures to roadmap research in question & answering (q&a). Technical report. www-nlpir.nist.gov/projects/duc/papers/qa.Roadmap-paper_v2.doc.

Gaizauskas, Robert, Mark A. Greenwood, Mark Hepple, Henk Harkemaa, Horacio Saggion, and Atheesh Sanka. 2005. The University of Sheffield's TREC 2005 Q&A Experiments. In *Proceedings of the 14th Text REtrieval Conference*.

Lavie, Alon and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, June. Association for Computational Linguistics.

Lin, Chin-Yew. 2004. Rouge: A package for automatic evaluation of summaries. In Marie-Francine Moens, Stan Szpakowicz, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

Manning, Christopher D. and Hinrich Schütze. 2001. *Foundations of Statistical Natural Language Processing*. MIT Press.

Voorhees, E. 2005. Overview of the TREC 2004 question answering track. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004)*. NIST Special Publication 500-261.