PKUTM Participation at TAC 2010 RTE and Summarization Track

Houping Jia*, Xiaojiang Huang*, Tengfei Ma, Xiaojun Wan, Jianguo Xiao

Institute of Computer Science and Technology

The MOE Key Laboratory of Computational Linguistics

Peking University, Beijing 100871, China

{jiahouping, huangxiaojiang, matengfei, wanxiaojun, xiaojianguo}@icst.pku.edu.cn

Abstract

This paper describes the systems of PKUTM in Text Analysis Conference (TAC) 2010. We participated in the Recognizing Textual Entailment (RTE) track and the Summarization track. For the RTE track, we propose a method to map every node in the hypothesis to one or more nodes in the text. With the help of named-entities tools, MINIPAR relationships, and regular patterns to recognize temporal and numeric expressions, some nodes are merged into one node. We transform the hypothesis by using semantic knowledge from sources like WordNet, VerbOcean, and LingPipe. In the Summarization track, we propose a unified framework for both kinds of summarization. We employ a manifold-ranking model to select sentences and a novel sentence ordering method to generate final summaries. The underlying idea of the proposed approach is that a good summary is expected to include the sentences with both high biased information richness and high information novelty. The evaluation results show that our proposed two frameworks are very effective for RTE and Summarization tasks, respectively.

1 Recognizing Textual Entailment Track

The TAC 2010 Recognizing Textual Entailment (RTE6) Main Task is similar to the RTE5 Search Pilot Task, which aims to find sentences in a collection of documents that logically entail particular "hypothesis" sentences. RTE6 does not include the traditional RTE Main Task as in the last five RTE challenges. There is no task to make entailment judgments over isolated T-H pairs drawn from multiple applications. Instead, the Main Task for RTE6 is based on only the Summarization application setting.

The difficulty of the task is threefold. First, the texts and hypotheses are not modified as compared to the original source, so they may contain incomplete sentences, spelling errors, grammar errors and abbreviations, etc. Second, texts and hypotheses are interpreted within the context of the topic, as they rely on explicit and implicit references to entities, dates, places, events, etc. pertaining to the corpus. Third, there are much more negative pairs than positive pairs, as for RTE6 DEVSET there are totally 15955 candidate pairs, while the number of positive entailment pairs is only 897.

1.1 System Overview

For both Main and Novelty tasks we use the same RTE engine. Our system applies transformations over the dependency-tree using a knowledge base of diverse types of entailment rules [Herrera et al.,

*Co-first authors.

2005; Iftene and Moruz, 2009]. For the positive pairs, we believe that the hypothesis can be transferred to the text by some transformation rules. For the negative pairs, we believe that there must be some mismatches between texts and hypotheses. The proposed system is based on surface techniques of lexical and syntactic analysis using knowledge from sources such as WordNet, VerbOcean, and LingPipe.

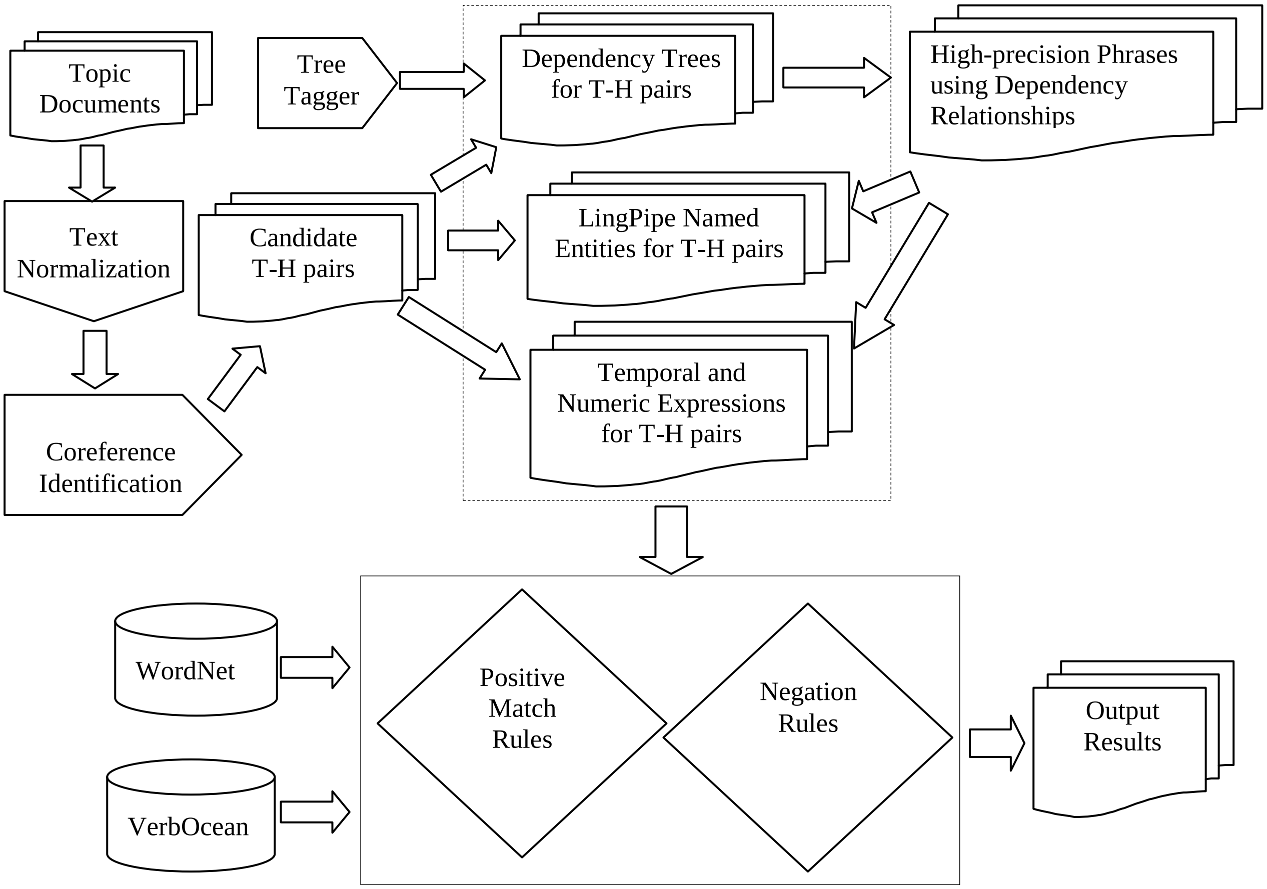
Topic Documents
Tree Tagger
Dependency Trees for T-H pairs
High-precision Phrases using Dependency Relationships
Text Normalization
Candidate T-H pairs
LingPipe Named Entities for T-H pairs
Temporal and Numeric Expressions for T-H pairs
Coreference Identification
WordNet
VerbOcean
Positive Match Rules
Negation Rules
Output Results

Figure1. RTE framework

We dealt with the task as the framework shown in Figure 1. The rest of this paper is organized as follows. In Section 1.2 we first describe text normalization, coreference identification within a document. After that, sentences and hypotheses within a topic are paired according to candidate information given by Lucene. Then we describe dependency analysis, named entity identification, and temporal and numeric expressions identification of texts and hypotheses. In Section 1.3 we present lexical and semantic match between tree nodes using WordNet and VerbOcean, followed by a description of determination of two texts (Section 1.4). Description of the submitted system results are detailed in Section 1.5. Section 1.6 contains ablation tests.

1.2 Preprocessing

1.2.1 Text Normalization: The first step is to improve the quality of the tools' output. We replace "*hasn't*" with "*has not*", "*isn't*" with "*is not*" [Iftene and Moruz, 2009] within one document. We prune sentences that begin or end with a quotation mark, and replace paired punctuation signs whose other half is missing with a white space. Sometimes, a sentence contains a newline character by mistake, e.g.

> *Defense arguments herald beginning of end to lengthy Air India*
> *terrorist bombing case.*

We replace the newline character ‘*n*’ with an empty character. The meaning of the text remains the same, but the output of MINIPAR and LingPipe is much better.

1.2.2 Coreference Identification: after the normalization step, the sentences within a document are sent to LingPipe coreference[1] to identify which entity mentions refer to the same entity. In the RTE6 tasks, sentences are situated within a set of documents. They rely on other sentences for their interpretation and their entailment is therefore dependent on other sentences as well. Hence, document coreference plays a crucial role in the inference process. The LingPipe coreference tool can only deal with pronouns, thus we apply some complementary methods to identify coreferring phrases after named entity identification [Mirkin et al., 2009].

1.2.3 Dependency Analysis: After coreference identification, sentences and hypotheses within a topic are paired according to candidate information given by Lucene. We parse the text and the hypothesis with MINIPAR [Lin, 1998], and use the TreeTagger tool[2] to replace the incorrect POS and lemma identified by MINIPAR [Iftene and Moruz, 2009]. In some cases, LingPipe named entity tool may miss, or recognize incorrect named entities. To address this problem, we have selected some useful MINIPAR relationships with high-precision to help find missing named entities and phrases, or to replace incorrect entities. The following relationships are used: “title”, “person”, “lex-mod”, “nn”, “amount-value”, and “num-mod”. With the “title” relationship, we can identify some persons, e.g. “*Dr. David Johnson*”, “*Sen. Arlen Specter*”, and “*President George W. Bush*”. We can also know “*President*” is the title, “*George*” is the first name, “*W.*” is the middle name and “*Bush*” is the last name. With the help of “abbrev”, “*FEMA*” is the abbreviation for the “*Federal Emergency Management Agency*”. “Amount-value” and “num-mod” help to identify some numeric expressions, such as “*about 65 kilometers*”, and we can also know “*kilometers*” is the “*unit measure*”, “*about*” is the quantification expression.

1.2.4 Named Entity Identification: The named entity recognizer in LingPipe extracts mentions of people, locations or organizations in English news texts. In the case of named entities of type PERSON, we additionally use the MINIPAR relationships so that “title” and “name” can be distinguished, and also “first name” and “last name” can be distinguished. For example, LingPipe only identifies the name “*David Johnson*” of one person “*Dr. David Johnson*” while MINIPAR can help find the title “*Dr.*” and distinguish first name and last name, thus whether it is a single title “*doctor*” or an omitted name “*Johnson*” or “*David*” who appears in the same document, it can be easily recognized as the same person. Also with the combination of MINIPAR, some entities can be corrected, e.g., LingPipe identifies “*George W. Bush*” as two PERSONs by mistake, “*George W.*” and “*Bush*”, and it recognizes the “*Irish Republican Army*” as one “PERSON” (“*Irish*”) and one “ORGANIZATION” (“*Army*”), while MINIPAR correct them as one semantic unit.

[1] http://alias-i.com/lingpipe
[2] http://www.cele.nottingham.ac.uk/~ccztk/treetagger.php

1.2.5 Numeric and Temporal Expressions Identification: We build a set of regular expression patterns to identify numeric and temporal expressions according to RTE past data sets and RTE6 development data sets. In addition, we choose a list of quantification expressions which are very important to numeric expression mapping, for example: "*about*", "*nearly*", "*as many as*", "*over*", "*or more*", "*other*", "*more than*", "*greater than*", "*lower than*", "*less than*", "*smaller than*", etc. Unit measures and quantification with number values are considered as a semantic unit with the help of MINIPAR's dependency relationships. Our system has the ability to identify years, dates, ages, measure values, money, etc. The followings are some examples from the RTE6 DEVSET:

Years: *in 2009, in the 1960s, in the late 1960s, late 1980s, late-1980s*

Dates: *Sept. 11, 2001, Sept 11, Sept. 11, September 11*

Ages: *78-year-old, 78*

Time Periods: *from 1965 to 1968, past 22 years, since 1950, four years ago*

Other Time: *5:30 p.m., two-hour, Thursday, Sunday afternoon, last month, tonight*

Measure Values: *More than 105 million Vioxx prescriptions, less than 25 milligrams, 25mg, about 30 yards, 1/2-foot-tall,50,000-acre, 40 miles*

Money: *$208,000, $2.5 billion, 2.5 billion dollars, 552.6-million-dollar, 48 euro cents, euro438.4*

Percentages: *18 percent*

Other numbers: *20/20, first, second, third-quarter, 17th*

However, we are not able to recognize all instances of numbers or time expressions using regular expression patterns, e.g. we have missed "*1/2-foot-tall*", "*20/20*" and "*50,000-acre*". Fortunately, MINIPAR's relationships have helped to find some of them, and to tell what the number value is and whether they have a unit measure or quantification.

1.3 Hypothesis Tree Transformation

Presently, the core of our approach is based on a tree node overlap algorithm applied on the dependency trees from hypothesis to text. The MINIPAR generates a dependency tree for every text and hypothesis. A module of lexical entailment is applied over the nodes of both text and hypothesis. Nodes which are recognized as phrases by MINIPAR, or named entities, numeric and temporal expressions are merged as one node. This is very important, for there are many incomplete sentences or sentences with abbreviations or omitted phrases in the documents, thus there are no mappings for many single words that are, in fact, part of larger semantic units. For a simple example, the T-H pair

> **T:** *The* ***memorial*** *is scheduled to open in 2009.*
> **H:** *The* ***World Trade Center Memorial*** *is due to open in 2009.*

is a true entailment pair. Here, if we do not consider "*World Trade Center Memorial*" as a whole in H which can be mapped to "*memorial*" in T, we will not find any mapping nodes for "*World*", "*Trade*", and "*Center*", thus the tree node overlap will decrease greatly.

For every node in the hypothesis tree which cannot be mapped morphologically to a node in the text tree, we have the following possibilities:

- If the word is a noun (or noun phrase), verb (or verb phrase), or adjective in the hypothesis tree, we use the WordNet [Fellbaum, 1998] relationships to lookup synonyms, hypernyms, holonyms, pertainyms for this word. For the verb nodes, we also use the VerbOcean relationships.
- If the word is marked as named entity by LingPipe, or if the word is a number or date time, we try to obtain information related to it from the background knowledge. In the event that even after these operations we cannot map the word from the hypothesis tree to one node from the text tree, we decide the final result: No entailment.

1.3.1 Morphology Match

If two nodes are morphologically equal, they are matched. Otherwise, we use the Levenshtein distance [Levenshtein, 1966] to compute the similarity. Two nodes s_1 and s_2 are matched if their Levenshtein distance *d* satisfies the following function.

$$\begin{cases} d <= 3, & \text{if } length(s_1) > 8 \wedge length(s_2) > 8 \\ d <= 2, & \text{if } length(s_1) > 5 \wedge length(s_2) > 5 \\ d <= 1, & \text{otherwise} \end{cases} \qquad (1)$$

1.3.2 WordNet and VerbOcean Based Match

If two nodes *t* and *h* are not morphologically matched, we consider their WordNet relations and their POS. We use a wide range of WordNet relations: synonyms, hypernyms, holonyms, meronyms, pertainyms, entailment, cause, etc. When *t* is a Verb and *h* is a Verb, we label the node pair as Verb-Verb. Different POS pairs have different match rules using WordNet relations. We consider six cases in total, Verb-Verb, Verb-Noun, Noun-Verb, Noun-Adjective, Adjective-Noun, Adjective-Adjective. These are the most popular pairs in the development samples.

Noun-Verb stands for the case where *t* is a noun and *h* is a verb, e.g. "*withdrawal*"-"*withdraw*" and "*diagnosis*"-"*diagnose*". In such case, if *t* matches *h,* the following rules should be satisfied:

any of *h*'s derived forms contains *t*,

or any of *t*'s derived forms contains *h*.

If *t* is a verb and *h* is a verb, we consider many more relations. In such case, we label *t*'s synonyms, derived forms, entailment words, meronyms, and holonyms as set S_t. We also collect *h*'s synonyms and hypernyms as S_h. If $h \in S_t \vee t \in S_h$, *t* matches *h*. For the adjective cases, we also consider specific relations, such as pertainyms, e.g. noun "*congress*" is a pertainym of adjective "*congressional*". There are some common words which are not taken into consideration, e.g. "new", "first" for adjectives.

VerbOcean [Chklovski and Pantel, 2004] is a broad-coverage semantic network of verbs. For the Verb-Verb pairs, we also consider all relations in VerbOcean except "opposite-of", e.g. "*strike*" is "similar" to "*attack*". Some common verbs are removed from VerbOcean which can bring about much noise, e.g. "*have*", "*do*", "*get*", "*need*", etc.

We do not apply the same match rules for all POS pairs, because it can compensate for missing pairs resulting in an increased overall system recall, but greatly decreasing the precision. All those rules are proved to be the most effective in the past RTE datasets. In order to improve the matching recall, we have considered some common suffixes, such as “s”, “er”, “or”, “ship”, “ing” for nouns and “ed”, “en” for verbs. For example, “*vacationing*” and “*holiday*” cannot be directly matched through Noun-Noun rules, but if we consider the suffix “ing”, we can map “*vacationing*” with “*holiday*” for “*vacation*” is a synonym of “*holiday*”.

Also, we add several pairs as a world knowledge base, such as “*money*”-“*fund*”, or “*gov*”-“*governor*” which can be matched through a bilingual dictionary. This knowledge base works although it is quite preliminary, and we will try to find better solutions in the later work.

1.3.3 Phrase Match

Phrases include named entities, temporal and numeric expressions and some other verb and noun phrases. For some simple phrases, we can use WordNet based match rules, like “*Irish Republican Army*” and “*take place*”. “*IRA*” is a synonymy of “*Irish Republican Army*” and “*happen*” is a hypernym of “*take place*”.

For temporal and numeric expressions, we define some rules using background knowledge. Some special situations need to be taken into account. Different numbers with different quantifiers can be synonymous. e.g. “*30 years*” is equal to “*three decades*”, and “*at least 30 people*” is implied by “*35 people*”. There are cases in which, even if the numbers are the same, certain unit measures or quantifiers may change their meaning for a negative match, for instance, “*at least 30 people*” is different from “*30 minutes*”.

For the named entities, we have defined different mapping rules for different types. Key terms or other types of prominent information that appear in the title or the first few sentences are often perceived as “globally” known throughout the documents [Mirkin et al., 2009]. For example, the geographic location mentioned at the beginning of the document is assumed to be known from that point on. Later reference of that location is usually abbreviated, e.g., “*foundation*” in the text has high probability to refer to the full expression “*World Trade Center Memorial Foundation*” in the hypothesis. For type PERSON, we add some commonly used titles like “*King*”, “*Queen*”, “*Senator*” in case the NER may miss. If only the title “*President*” appears in a text, and the name with corresponding title “*President George W. B*ush” is in the hypothesis, there is great possibility that they point to the same person. We also consider “*Dr. David Johnson*” and “*Mr. Johnson*” referring to the same person although they have different surface titles, and that “*Specter*” refers to “*Sen. Arlen Specter*”, despite the omitted name and title.

1.3.4 Negation and Antonymy

Negation is detected when a node is found in a negation relationship with its father in the dependency tree. The negation relationship is then propagated to its ancestors all the way to the head. The entailment between nodes affected by negation is implemented based on the antonymy relation of WordNet, the “opposite-of” relation in VerbOcean and negation words, e.g. “*no*”, “*not*”, “*nobody*”, “*without*”.

If two nodes *h* and *t* are matched according to previous rules and one of them is negated, their entailment relation is false. If two nodes *h* and *t* have an antonymy relation and one of them is negated, they have a positive entailment relation. Some verbs such as “*find*”, “*live*”, “*send*” are excluded in this rule because that bring much noise.

1.4 Determination of Entailment

Dependency trees give a structured representation for every text and hypothesis. Mapping between dependency trees can give an idea about how semantically similar two text snippets are. For every node from the hypothesis tree, we calculate the matching value and afterwards consider the normalized value relative to the number of nodes from the hypothesis tree. A stop list[3] with 571 terms is applied to remove frequent token nodes.

A higher degree of matching between dependency trees has been taken as indication of a semantic relation. There are much more negative pairs than positive pairs, so the threshold is essential to determine whether there is an entailment relation between a text and a hypothesis. In our experiments, we have obtained the threshold after training the system with the development corpus.

1.5 Results

We submit three results for Main Task and Novelty Task. The best submission PKUTM2 employs the whole procedure described in the previous section. The PKUTM1 has removed the LingPipe coreference module from PKUTM2, and the PKUTM3 has removed the world knowledge base from PKUTM2.

	Precision (%)	**Recall (%)**	**Micro-averaged F1 (%)**
PKUTM1	70.14	36.30	47.84
PKUTM2	68.57	36.93	**48.01**
PKUTM3	68.69	35.98	47.22

Table 1. RTE6 Main Task Evaluation Results

	Evaluation Micro-averaged F1 (%)	**Justification Micro-averaged F1 (%)**
PKUTM1	**82.91**	47.88
PKUTM2	82.76	**48.26**
PKUTM3	82.55	47.66

Table 2. RTE6 Novelty Task Evaluation Results

We analyze T-H pairs from the development set, and find that most of the negative pairs have very low node overlap, and there are much fewer negative pairs with high overlap than positive pairs with high overlap. Hence, we set a high threshold in order to keep high precision although we may miss a lot of positive pairs. The results in Table 1 show that we got high precision values but low recall values as expected.

In order to improve recall, we have to distinguish between positive and negative cases in which text and hypothesis contain matched words between too many and too few. The solution for this problem is to use a special tool that identifies semantic roles for words and to apply new rules for cases in

[3] http://www.lextek.com/manuals/onix/stopwords2.html

which the matched word has different roles in text and in hypothesis.

1.6 Ablation Tests

To be able to see each component's relevance, we have submitted three ablation tests for Main Task. The results in Table 3 show that the system's rules related to named entities are the most important, and VerbOcean and Coreference also have some positive contribution.

	Precision (%)	**Recall (%)**	**Micro-averaged F1 (%)**	**Relevance (%)**
Without LingPipe Coreference	70.14	36.30	47.84	0.38
Without VerbOcean	69.28	35.56	46.99	1.27
Without LingPipe NER	80.39	21.69	34.17	**13.84**

Table 3. RTE6 Main Task Ablation Results

Since texts and hypotheses rely on explicit and implicit references to entities, dates, places, events, etc. pertaining to the corpus, we have used effective rules related to named entities. However, we do not consider the "events", as the following example:

T: *The* ***IRA*** *rejected* ***those terms*** *in December and since has been implicated in a string of criminal scandals, including a world-record bank robbery, the knife slaying of a Catholic man and a money-laundering network.*

H: *The* ***Irish Republican Army*** *refused to* ***produce photographic evidence that its arms had been destroyed****.*

If we know that the event "*those terms*" refers to "*produce photographic evidence that its arms had been destroyed*" in the context, we can make the right decision.

2 Summarization Track

The TAC 2010 Guided Summarization Task aims to encourage summarization systems to make a deeper linguistic analysis of the source documents to generate short fluent multi-document summaries. For a given topic, all the documents are separated into two document sets Set A and Set B. Systems are required to generate an initial summary of documents in Set A, and a update summary of documents in Set B with the assumption that the documents in Set A have been read.

We propose a unified framework for both kinds of summarization. The main difference lies in the "sentence selection" step, which will be discussed in Section 2.3.

2.1 System Overview

The system architecture is shown in Figure 2. In this framework, we apply a manifold-ranking model to select sentences for summaries. After the ranking process, we propose a novel sentence ordering method to generate final summaries.

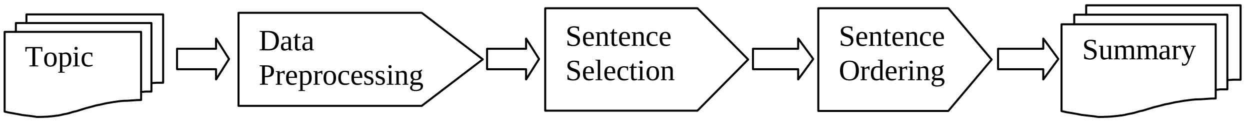

Figure 2. Summarization framework

2.2 Guided Summarization

2.2.1 Data Preprocessing

The document preparation step begins with content extracting. We split documents into sentences, and tokenize sentences into words. Sentences which are too short (shorter than three non-stop words) are eliminated. After that, a named entity recognizer is applied to extract mentions of people, locations or organizations in each sentence.

2.2.2 Sentence Selection

In this step, we apply an approach based on manifold-ranking [Wan et al., 2007] of sentences to topic-focused multi-document summarization. The manifold-ranking based summarization approach consists of two steps: (1) the manifold-ranking score is computed for each sentence in the manifold-ranking process where the score denotes the biased information richness of a sentence; (2) based on the manifold-ranking scores, the diversity penalty is imposed on each sentence and the overall ranking score of each sentence is obtained to reflect both the biased information richness and the information novelty of the sentence. The sentences with high overall ranking scores are chosen for the summary.

2.2.2.1 Manifold-Ranking Process

The manifold-ranking method is a universal ranking algorithm and it is initially used to rank data points along their underlying manifold structure. The prior assumption of manifold-ranking is: (1) nearby points are likely to have the same ranking scores; (2) points on the same structure (typically referred to as a cluster or a manifold) are likely to have the same ranking scores. An intuitive description of manifold-ranking is as follows: A weighted network is formed on the data, and a positive rank score is assigned to each known relevant point and zero to the remaining points which are to be ranked. All points then spread their ranking score to their nearby neighbors via the weighted network. The spread process is repeated until a global stable state is achieved, and all points obtain their final ranking scores.

The details of the manifold-ranking algorithm can be found in [Wan et al., 2007], and there are some changes in our experiments as follows:

1) In the first step, we apply an asymmetric measure to compute the pair-wise similarity values between sentences besides the standard Cosine measure. The standard Cosine symmetric similarity computed as the normalized inner product of the corresponding term vectors, and the asymmetric measure is computed as the inner product divided by the smaller vector modulus.

2) We set the initial ranking score as

$$f^0(i) = \begin{cases} \alpha, & \text{if } i = 0 \\ (1-\alpha) n_e(i) / N_e, & \text{otherwise} \end{cases} \tag{2}$$

where n_e denotes the number of named entities each sentence contains and N_e is the total number of named entities.

2.2.2.2 Diversity Penalty Imposition:

There are some sentences which are not suitable for summary because of their form of expression, such as questions, direct quotations, very short sentences, etc. We use a few regular expressions to remove such sentences. Sentences shorter than five words, or having unmatched punctuations, or containing “say” verbs, e.g. “said”, “says”, “tell”, “told” , are excluded from final summaries.

The details of the diversity penalty algorithm can also be found in [Wan et al., 2007], and we have changed the end of iterative condition (step 4): Go to step 2 and iterate until $B = \phi$ or the length of sentences in A reaches a predefined maximum number (100 words).

After the overall ranking scores are obtained for all sentences, several sentences with the highest ranking scores are chosen to make up the summary according to the summary length limit.

2.2.3 Sentence Ordering

A sentence ordering method is proposed in order to improve the readability and fluency of final summaries. Our motivation is to preserve the original orders of sentences in the documents as much as possible. So we order the summary sentences by the order of their projections in the documents. Suppose we have the selected sentences $R = \{s_i \mid 1 \le i \le k\}$ from documents $D = \{d_i \mid 1 \le i \le k\}$, the ordering process consists of three steps as follows:

1. For each sentence s_i, find a sequence of sentences $\{m_{ij}\}$, where m_{ij} has the maximum similarity sim_{ij} with s_i in document d_j. The similarity is estimated by the standard Cosine measure.

2. Compare the order of two sentences s_i and s_j according to the following formulations.

$$order(s_i, s_j) = sign\left(\sum_d I(s_i, s_j, d)\right) \tag{3}$$

$$I(s_i, s_j, d) = \begin{cases} sign\left(id(m_{i,d}) - id(m_{j,d})\right), & \text{if } (s_i \in d \vee s_j \in d) \wedge sim_{i,d} > \tau \wedge sim_{j,d} > \tau \\ 2 \times sign\left(id(m_{i,d}) - id(m_{j,d})\right), & \text{if } s_i \notin d \wedge s_j \notin d \wedge sim_{i,d} > \tau \wedge sim_{j,d} > \tau \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$sign(x) = \begin{cases} 1, & x \ge 0 \\ -1, & x < 0 \end{cases} \tag{5}$$

3. Rank all sentences according to the order of two sentences. Assuming that $\{s_{t1}, s_{t2} \ldots s_{tm}\}$ is the set of sentences already ranked, we insert sentence s_{m+1} from the back of the sequence until $order(s_{m+1}, s_{t1}) > 0$ and $order(s_{m+1}, s_{t1-1}) < 0$.

2.3 Update Guided Summarization

The update summarization has the same framework as initial summarization except for some differences in the sentence selection process. Since we are allowed to use the prior knowledge from Set A while dealing with the content of Set B documents, we add the summary sentences of Set A to the sentence network at the beginning of manifold-ranking process and remove them at the beginning of sentence selecting. Suppose we have the manifold-ranking score $f(x_i)$ for each sentence x_i in Set B. In the first step of diversity penalty algorithm, the overall ranking score is not initialized to the manifold-ranking score. Instead, we initialize the score of each sentence in Set B as follows:

$$RankScore(x_j) = f(x_j) - \sum_{x_i \in Summ_A} \omega \cdot \bar{S}_{ji} \cdot f\left(x_i\right), \quad \forall x_j \in B \tag{6}$$

2.4 Results

NIST assessors wrote 4 model summaries for each document set. All submitted systems are evaluated both automatically and manually, including ROUGE-2, ROUGE-SU4, Pyramid, Linguistic Quality and Overall Responsiveness. We submit two runs for our system. PKUTM1 uses the standard symmetric Cosine Similarity directly while PKUTM2 uses the asymmetric Similarity in the first step of manifold ranking algorithm. Table 4 and table 5 shows the performance of systems in initial summarization and update summarization respectively. The Model row shows the average results of manually written summaries are, and the highest results of participants in each metric are reported under the name BestSystem.

	ROUGE-2	**ROUGE-SU4**	**Pyramid**	**Linguistic Quality**	**Overall Responsiveness**
Model	0.115	0.152	0.785	4.908	4.761
PKUTM1	0.085	0.120	0.386	3.283	3.022
PKUTM2	0.085	0.119	0.375	3.043	2.978
BestSystem	0.096	0.130	0.425	3.652	3.174

Table 4. Initial Summarization Results

	ROUGE-2	**ROUGE-SU4**	**Pyramid**	**Linguistic Quality**	**Overall Responsiveness**
Model	0.097	0.134	0.673	4.821	4.712
PKUTM1	0.066	0.108	0.247	2.935	2.370
PKUTM2	0.071	0.110	0.243	2.848	2.370
BestSystem	0.080	0.120	0.321	3.739	2.717

Table 5. Update Summarization Results

The evaluation results show that even best automatic summarization system cannot compare with human brains in all metrics. The linguistic qualities of most systems are not bad, because the sentences are extracted from ordinary documents without major modification. Although the ROUGE scores of models and all systems are low, they are correlative to the overall responsiveness. Thus the ROUGE is a good metric for automatic evaluation. Our system PKUTM1 achieves the 5th place in the initial summarization and 12th place in the update summarization among the 23 participants and 43 runs according to the overall responsiveness metric. The PKUTM2 achieves the 10th and 13th place in the initial and update summarization respectively, which is slightly worse than the PKUTM1.

3 Conclusion

These systems mark PKUTM's first participation in an organized evaluation for RTE and Summarization. For the RTE system, we propose a method, to map every node in the hypothesis to one or more node in the text. We will focus on semantic roles for words to improve recall and dependency relationships to improve precision for further study. For the Summarization task, we apply a manifold-ranking model to select sentences and a novel sentence ordering method to generate final summaries. From evaluation results, we can see that our systems have achieved competitive results.

4 Acknowledgments

This work was supported by NSFC (60873155), Beijing Nova Program (2008B03) and NCET (NCET-08-0006). We thank Feng Liang and Shanshan Huang for proofreading the paper.

References

[Chklovski and Pantel, 2004] T. Chklovski and P. Pantel. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. *In Proceedings of EMNLP 2004.*

[Fellbaum, 1998] C. Fellbaum. Wordnet: An electronic lexical database. MIT Press, Cambridge.

[Herrera et al., 2005] J. Herrera, A Pe ñas, and F. Verdejo. Textual Entailment Recognition Based on Dependency Analysis and WordNet. I*n Proceedings of PASCAL RTE 2005.*

[Iftene and Moruz, 2009] A. Iftene and M.-A. Moruz. UAIC Participation at RTE5. *In Proceedings of TAC 2009.*

[Levenshtein, 1966] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *In Soviet Physics - Doklady, volume 10, pages 707-710, 1966.*

[Lin, 2008] D. Lin. 1998. Dependency-based Evaluation of MINIPAR. *In Workshop on the Evaluation of Parsing Systems at ICLRE 1998.*

[Mirkin et al., 2009] S. Mirkin, R. Bar-Haim, J. Berant, I. Dagan, E. Shnarch, A. Stern, and I. Szpektor. Addressing discourse and document structure in the RTE search task. *In Proceedings of TAC 2009.*

[Wan et al., 2007] X. Wan, J. Yang, and J. Xiao. Manifold-ranking based topic-focused multi-document summarization. *In Proceedings of IJCAI 2007.*

[Zhang et al., 2005] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W.-Y. Ma. Improving web search results using affinity graph. I*n Proceedings of SIGIR'2005.*