Weakly Supervised Natural Language Processing Framework for Abstractive Multi-Document Summarization

Peng Li
Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, 76019
jerryli1981@gmail.com

Weidong Cai
School of Information Technologies
University of Sydney
NSW 2006, Australia
tom.cai@sydney.edu.au

Heng Huang [*]
Computer Science and Engineering
University of Texas at Arlington
Arlington, TX, 76019
heng@uta.edu

ABSTRACT

In this paper, we propose a new weakly supervised abstractive news summarization framework using pattern based approaches. Our system first generates meaningful patterns from sentences. Then, in order to precisely cluster patterns, we propose a novel semi-supervised pattern learning algorithm that leverages a hand-crafted list of topic-relevant keywords, which are the only weakly supervised information used by our framework to generate aspect-oriented summarization. After that, our system generates new patterns by fusing existing patterns and selecting top ranked new patterns via the recurrent neural network language model. Finally, we introduce a new pattern based surface realization algorithm to generate abstractive summaries. Automatic and manual evaluations demonstrate the effectiveness and advantages of our new methods. Code is available at: https://github.com/jerryli1981

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Text analysis

General Terms

Algorithms, Experimentation

Keywords

Summarization; Recurrent Neural Network; Capped Norm Semi-Supervised Learning

1. INTRODUCTION

Multi-document summarization (MDS) problem has been mainly solved by the sentence extraction and compression based approaches [12, 4, 8, 13] for decades. However, the top systems so far were only considered as barely acceptable by the human assessment. Moreover, the empirical study [6] found the performance ceiling of pure sentence extraction approaches, which is much lower than human summaries and not much better than the current best automatic systems. Meanwhile, user behavior experiments conducted independently by [6, 20] demonstrated that human always do abstraction instead of simply selecting relevant sentences. More specifically, human tempt to interpret the text to understandable representation in their mind. After that, such representation is transformed to summary representation, also known as content selection. Finally, human generate summary text from summary representation [11]. Therefore, based on human cognitive process, we believe that a fully abstractive approach has the most potential for generating summaries at the level comparable to human summary.

There are some pioneer works towards abstractive summarization (see Section 2). For example, in meeting conversation analysis community, some pattern based approaches first extract patterns from human-authored meeting summaries. Then, their approaches select relevant patterns via clustering and ranking. Finally, the template filling is used to generate summary text. In this paper, we focus on a new and distinct problem – how to generate news events summaries using pattern based abstractive summarization framework. Unlike abstractive meeting conversation summarization, abstractive news summarization faces several unique challenges. First of all, there exist several informative long sentences that have complex syntax structure in each news corpus. Secondly, the approaches of meeting conversation summarization can leverage domain information such as dialogue act or meeting participants to help generate abstractive summaries. However, domain information in news events are not easy to be captured. Thirdly, the standard text generation technologies such as template filling may lead to worse readability in news summarization scenario.

To solve these challenges and achieve the goal of generating high quality abstractive summarization, we propose a new abstractive summarization framework which includes an enhanced pattern extraction module via integrating advanced information extraction technology, a novel robust semi-supervised pattern learning algorithm, and a novel surface realization algorithm for generating abstractive news summaries. More specifically, we first extract tuples by OpenIE [1], recognize argument and predicate's head via Stanford CoreNLP [2], and annotate the type of phrase head via SEMAFOR [3] and wordnet *etc*. Secondly, in order to precisely cluster patterns that better describe user interested information such as what happened, reason, damages and countermeasures, we propose a novel semi-supervised pattern learning algorithm based on a new capped norm based objective function, which can achieve more optimal semi-supervised learning results than existing approaches. Thirdly,

[*] Corresponding author.

[1] http://openie.cs.washington.edu

[2] http://nlp.stanford.edu/software/corenlp.shtml

[3] http://www.ark.cs.cmu.edu/SEMAFOR/

CIKM'15, October 19–23, 2015, Melbourne, Australia.
© 2015 ACM. ISBN 978-1-4503-3794-6/15/10 ...$15.00.
DOI: http://dx.doi.org/10.1145/2806416.2806494.

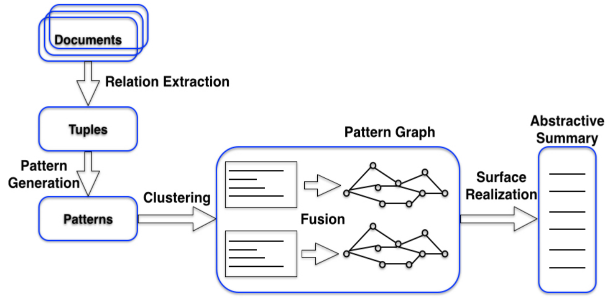

Figure 1: Abstractive news summarization framework flowchart

our system generate new patterns by fusing learned patterns and select top ranked new patterns and sentences via recurrent neural network language model. Finally, we use Integer Linear programming to select top ranked sentences cross clusters. Figure 1 shows the flowchart of our new abstractive summarization framework.

The key contributions in our work are summarized as follows:

1) The first weakly supervised abstractive news summarization system that only leverages a small number of hand-crafted keywords. Note that interpretation and generation phases of our system are purely unsupervised. The reason for us to use keywords for helping clustering patterns is that we hope to generate aspect-oriented summarization [4]. The goal of aspect-oriented summarization is to present the most important content to the user in a condensed form and a well-organized structure to satisfy the user's needs. For example, a summary about "Attacks" event should include aspects about what happened, when/where it happened, reasons, damages, rescue efforts, *etc*. Table 1 shows the aspects and keywords for the "Attacks" event. We use keywords to accomplish domain-specific abstractive summarization.

2) In order to improve pattern clustering purity, we make use of continuous word vectors derived from a recurrent neural network architecture [19]. The vector-space word representations learned from the language models were shown to capture syntactic and semantic regularities. The word relationships are characterized by vector offsets, where in the embedded space, all pairs of words sharing a particular relation are related by the same constant offset. Considering that this distributional semantic theory may benefit our pattern clustering component, we leverage word representations trained from large external data to differentiate patterns.

3) To effectively utilize hand-crafted keywords, we propose a new semi-supervised pattern learning model to group the patterns. Because the number of labeled data is very small, the supervised learning models and many semi-supervised learning (*e.g.* transductive support vector machine) methods cannot get good performance on the pattern labeling. Thus, we use the label propagation based semi-supervised learning model. The previous label propagation based semi-supervise learning models use the ℓ_2-norm loss, which cannot achieve the ideal semi-supervised learning results. Although the ℓ_0-norm based objective is desired in semi-supervised learning, it is an NP-hard problem. To reach the optimal labeling results, we propose a new capped norm based objective, which leads to more optimal results than traditional methods. We also derived a new and concise optimization algorithm to solve the proposed non-smooth and non-convex objective with proved convergence.

4) In order to control wrong facts and generate meaningful sentences given high quality pattern template, our system first filters out noisy templates by estimating the grammaticality with the recurrent neural network language model. After that, in order to generate meaningful sentences, we fuse tuples and select the best paths from the tuple graph that satisfies two ranking criteria. One criteria is the path should best match the most fluent pattern template, and the other criteria is the path should be the most fluent path compared with other paths. Finally, we use Integer Linear programming method to select top ranked facts cross clusters.

2. RELATED WORK

One line of related works focus on the task of news headline abstraction. Alfonseca *et al.* [1] proposed a framework to generate abstractive news headline. They trained a pattern clustering model via Noisy-OR Bayesian network and computed probability distribution of newly extracted patterns from testing news collection via two-step random walk on bayesian network. They selected the pattern with highest ranking score from the most representative pattern cluster as template to generate headline sentence. Pighin *et al.* [25] solved the same problem with the same framework. However, they found that the memory-based pattern extraction method is better than heuristic or sentence compression based pattern extraction.

Another line of research works focus on abstractive meeting conversation summarization. Murray *et al.* [20] followed the framework proposed by Jone [11] for generating abstractive summaries. In interpretation phase, they trained several classifiers to map sentences to domain dependent conversation ontology, then further mapped to predefined message patterns. After that, they utilized integer linear programming to select informative patterns in content selection phase. Finally, ontology based text plan and surface realization was implemented to generate abstractive summarization. Wang *et al.* [26] followed the same generation framework to solve meeting conversation summarization. In interpretation phase, they first clustered human-authored summary sentences and applied a multiple-sequence alignment algorithm to generate templates. In content selection phase, they identified the cluster describing a specific aspect and extracted all summary-worthy relation instances. In generation phase, the templates were filled with these relation instances, then found the top ranked instances to form final summary. Mehdad *et al.* [17] grouped similar sentences with supervised classifier and used the trained entailment graph to select representative sentences. After that, they built a word graph over similar sentences for fusion and selecting highly ranked paths as the final summaries. Oya *et al.* [24] extracted templates from clustered human-authored summaries and further got the generalization via template fusion in interpretation phase. In content selection phase, they borrowed training data to select the best templates. Finally, the templates were filled with matching labels to create summaries. Mehdad *et al.* [16] constructed word graph on human utterances and then selected the top ranked paths corresponding to the queries to generate the final summary.

Genest *et al.* [7] presented a fully abstractive news summarization approach. They pointed out a fully abstractive summarization framework should comprise analysis of the text, content selection and summary generation. However, they created event template manually with linguistic resources. Mausam *et al.* [15] automatically extracted relations from text without requiring a pre-specific vocabulary, by identifying relation phrases and associated arguments in arbitrary sentences. Following their work, Balasubramanian *et al.* [2] generated coherent event schemas. Zhang *et al.* [29] focused on Chinese poetry generation task. Their poetry generator jointly performs content selection and surface realization by learn-

[4] http://www.nist.gov/tac/2010/Summarization/

Aspects	Explanations	Hand-crafted keywords
WHAT	what happened	felony, lawlessness, assault, theft, rape, threat, *etc.*
WHEN	date, time, other temporal placement markers	January, Monday, today, yesterday, *etc.*
WHERE	physical location	nearby, street, road, highway, spot, blvd, *etc.*
PERPETRATORS	individuals or groups responsible for the attack	criminal, agent, offender, bomber, terrorist, *etc.*
WHY	reasons for the attack	motive, commit, intent, cause, means, reason, *etc.*
WHO AFFECTED	casualties (death, injury), or individuals otherwise negatively affected	casualty, death, loss, wound, hurt, missing, *etc.*
DAMAGES	damages caused by the attack	ruin, burn, property, possession, loss, *etc.*
COUNTERMEASURES	countermeasures, rescue efforts, prevention efforts, other reactions	prevent,, secure, surveillance, police, *etc.*

Table 1: Aspects for TAC 2010 guided summarization task, category 2: Attacks

ing representations of individual characters via the recurrent neural network.

3. PATTERN GENERATION

We observed that nearly half of the sentences comprise multiple aspects and complicated grammatical structures. In order to better interpret complicated sentences, we expect to extract meaningful relations and split structure-complicated sentences into simple ones. Table 2 illustrates the results of tuple extraction and pattern generation by our system.

In order to generate high quality patterns, we first utilize the sophisticated relation extraction tool such as openIE to generate tuples which have the triple form of {argument1, predicate, argument2}. After that, the head of arguments are replaced with their semantic types. There are two major challenges here. One is how to find argument head precisely if argument contains complicated phrase structure. The other is where to find type ontology and how to design type mapping rules effectively. To solve these challenges, we need linguistic resources and toolkits such as Framenet [5] and Stanford CoreNLP. To leverage Stanford CoreNLP annotation pipeline for pattern quality enhancement, we need map openIE annotation schema to Stanford CoreNLP annotation schema.

3.1 Preprocessing Tuples

Predicate in each relation tuple may share some common grammatical structures such as verb → noun → prep, verb, verb → prep, be→ verb, be→ verb→ prep, adv→ verb→ noun→ prep, be→ prep etc. However, we found OpenIE put prepositional token into argument 2. For example, {“he”, “stood two hours”, “after killing three of them”}. Therefore, when facing these special cases, we move the preposition into predicate. In order to locate prepositional word, two grammatical relations [6] are used with heuristic rules. One is “pobj” relation, the other is “pcomp” relation. After reassembling, the tuple may become {“he”, “stood two hours after”, “killing three of them”}.

3.2 Token Mapping

Before argument head extraction via Stanford CoreNLP toolkit, we need map string tokens in each tuple to Stanford IndexedWord objects. The mapping is not easy to implement because openIE and Stanford CoreNLP use different tokenization algorithm that may result in tokenization inconsistence. For example, Stanford CoreNLP treats “Us-led” as one token but openIE just considers “led” as tuple’s predicate. We therefore use the token offset as a bridge to conduct token mapping due to tokens in each tuple may not be continuous. More specifically, our system first generates relation tuples and core annotations including pos, lemma, NER, parse tree and semantic dependency graph, and stores token start offset to IndexedWord mappings. If the mapping relation can’t be found due to tokenization inconsistence, we use heuristic rules to match correct start offset. If original sentence doesn’t contain argument mention due to token order inconsistence, we search begin position in substring via the regular expression and ignore the isolated tokens. After mapping between tuple tokens and IndexedWord objects, we can fully make use of semantic dependencies between tuple tokens to extract phrase head.

3.3 Phrase Head Extraction

Argument and predicate head extraction module is quite important for later head type recognition and pattern clustering. The main challenge here is how to extract head without loosing readability. The arguments and predicate in each relation tuple may have complicated phrase structure. For example, {“The suspect”, “apparently called his wife saying”, “he was acting out in revenge for something”}, {“Kirchner”, “told The Associated Press”, “that six people were killed”}. The argument 2 of these two tuples both contain subsentence or clause. Oya *et al.* [24] treated the right most nouns as the head nouns, but this heuristic rule is not always true. Paper [2] used several parts of speech based heuristic rules to conduct argument head extraction. However, we found this approach didn’t work very well especially when facing such complicated phrase structures in tuples. We found phrase head extraction on complicated phrases may destroy tuple readability. Therefore, when facing argument mention contains grammatical relation like “nsubj”, “nsubjpass” or “ccomp”, we just ignore it. This kind of ignorance would make sense due to the fact that openIE will find {“six people”, “were”, “killed”} relation tuple occasionally.

After the above step, we conduct depth first search on parser tree with the help of CollinHeadFinder rules to locate phrase head. For those heads can’t be recognized by CollinHeadFinder, we design heuristic rules on semantic dependency graph to locate potential heads. For example, if grammatical relation is one of {“det”, “amod”, “poss”, “num”, “advmod” }, then the governor token is the head. If grammatical relation is one of {“tmod”, “nn”}, then the dependent is the head.

3.4 Head Type Tagging

After phrase head extraction, we construct patterns via replacing the heads with their types. Standford NER as the state of

[5] `https://framenet.icsi.berkeley.edu/fndrupal/`

[6] `http://nlp.stanford.edu/software/dependencies_manual.pdf`

Aspects	Tuple	Pattern	Simple Sentence
What happened	{“girl”, “killed by”, “gunman”}	{“PERSON”, “killed by”, “PROTAGONIST”}	Five girls killed by a gunman inside their one-room, tiny schoolhouse.
Countermeasures	{“buggies”, “carried”, “mourners”}	{“CONTAINER”, “carry”, “ACT”}	34 buggies and carriages carried mourners.
Where	{“girls”, “inside”, “schoolhouse”}	{“PERSION”, “inside”, “LOCATION”}	five girls inside tiny one-room schoolhouse.

Table 2: An example of splitting structure-complicated sentence into structure-simple sentences via tuples and patterns. Original sentence is “A procession of 34 buggies and carriages carried mourners to a hilltop cemetery Thursday as the Amish community buried the first of five girls killed by a gunman inside their tiny one-room schoolhouse.”

the art type tagging tool have good performance on a small number of types. However, we found it can’t recognize the tokens like “man”, “girl”, “chilldren” as person. Also we recognize that the SEMAFOR as the state-of-the-art semantic role labeling tool didn’t work well to recognize pronoun. In our case, some argument heads are not name entities, and they are concepts such as “school”, “man”. Therefore, a combination of several tools to conduct type tagging is a good choice.

Balasubramanian *et al.* [2] used wordnet and Stanford NER combination to conduct head type tagging. They manually selected 29 semantic types from WordNet. Because they built event schema in open domain, their types are too general for our task. Alfonseca *et al.* [1] used Wikipedia and Freebase to annotate head with all its Freebase types and got better results. However, their tagging tool is company private property, not for sharing. Inspired by Chen *et al.* [3] on slot filling, we use SEMAFOR semantic parser based on framenet combined with Stanford NER and wordnet to recognize argument head’s type. In our implementation, Stanford NER has priority to find person, location, time and date. For those heads can’t be recognized by Stanford NER, we search Framenet and try to match frame elements via SEMAFOR tool. Note that SEMAFOR may produce noisy when labeling. For example, Tuple {“he”, “picked up”, “milk”} will generate pattern like {“person”, “pick up”, “food”}, {“a truck”, “picked up”, “milk”} will generate pattern like {“vehicle”, “pick up”, “chosen”}. The same “milk” generates two different labels. Therefore, We use wordnet, DBpedia and freebase to filter wrong labeling.

4. PATTERN LABELING

After generating patterns, we use hand-crafted keywords to group aspect relevant patterns for later template generation. In order to precisely group patterns that better describe user interested event aspects such as what happened, reason, damages and countermeasures, we propose a novel semi-supervised pattern labeling algorithm.

4.1 Construct Pattern Feature Vector

We use word2vec [7] as the word embedding resources. It is generated by learning a recurrent neural network [19]. The recurrent neural network language models use the context history to include long-distance information. The word relationships are characterized by vector offsets, where in the embedded space, all pairs of words sharing a particular relation are related by the same constant offset. Considering that this distributional semantic theory may benefit our pattern clustering task, we leverage word representations trained from large external data to generate pattern feature vectors. Word2vec contains pre-trained vectors trained on part of Google News dataset(about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases.

Recall that our pattern has the triple form of {arg1Type, predicate, arg2Type}. Therefore, we concatenate the vectors of arg1Type, head of predicate and arg2Type to form pattern feature vector. On the other hand, we use merge strategy to construct labeled feature vectors of hand crafted keywords.

4.2 Capped Norm Based Semi-Supervised Pattern Learning Model

In order to accurately label the patterns via the hand crafted keywords, we propose a novel capped norm based semi-supervised pattern learning model. Given input $\{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$ where $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_c, \mathbf{x}_{c+1}, \cdots, \mathbf{x}_n\} \subset \Re^d$ is the pattern set, the first c elements are hand crafted keywords with labels as $\{\mathbf{y}_1, \ldots, \mathbf{y}_c\}$ where $\mathbf{y}_i$ is a label indicator vector of size c containing the labels assigned to the pattern $\mathbf{x}_i$: $\mathbf{y}_i(k) = 1$ if $\mathbf{x}_i$ belongs to the k-th class, and 0 otherwise. Our goal is to predict the label sets $\{\mathbf{y}_{c+1}, \cdots, \mathbf{y}_n\}$ for the unlabeled patterns $\{\mathbf{x}_{c+1}, \cdots, \mathbf{x}_n\}$.

Traditional graph based semi-supervised learning model solves the following problem [31, 30]:

$$\min_Q Tr(Q^T L Q) + \gamma Tr(Q - Y)^T (Q - Y), \qquad (1)$$

where L is the Laplacian matrix and $Q \in \Re^{n \times c}$ is the predicted decision values matrix to be solved. Given an affinity matrix W, the Laplacian is defined as $L = D - W$, where $D = \text{diag}(W\mathbf{e})$, $\mathbf{e} = (1, \ldots, 1)^T$. The way to learn affinity matrix will effect the final results, thus in our experiments we test two popularly used methods: heat kernel and cosine similarity.

We can further re-write problem (1) as:

$$\min_Q \sum_{i,j=1}^{n} W_{ij} ||\mathbf{q}^i - \mathbf{q}^j||_2^2 + \gamma Tr(Q - Y)^T (Q - Y). \qquad (2)$$

The ideal solution Q for semi-supervised learning is that $\mathbf{q}^i = \mathbf{q}^j$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class, which means many rows of Q are equal and thus has ideal class indicators. That is to say, it is desired that $||\mathbf{q}^i - \mathbf{q}^j||_0 = 0$ for many pairs (i, j) (the ℓ_0-norm of $\mathbf{w}$ is a count of the nonzero elements in $\mathbf{w}$). However, the ℓ_0-norm minimization problem is an NP-hard problem. Although many recent works used ℓ_1-norm to approximate ℓ_0-norm [23], the gap between them is still large. To solve this problem and obtain a more ideal solution Q, we propose to use the capped norm based loss function [27, 28, 10] and solve the following problem for semi-supervised learning:

$$\min_Q \sum_{i,j=1}^{n} W_{ij} \min\{||\mathbf{q}^i - \mathbf{q}^j||_2, \varepsilon\} + \gamma Tr(Q - Y)^T (Q - Y)\,, \qquad (3)$$

[7] https://code.google.com/p/word2vec

Algorithm 1 The algorithm to solve the proposed capped norm semi-supervised learning model as in (3).

```
**Input** The original weight matrix $W \in \Re^{n\times n}$. $D$ is a diagonal matrix with the $i$-th diagonal element as $\sum_j W_{ij}$. The initial label matrix $Y \in \Re^{n\times c}$.
$t = 1$. Initialize $Q_t \in \Re^{n\times c}$ using standard semi-supervised learning result as in Eq. (1);
**while** Not Converge **do**
    1. Calculate $\widetilde{W}_{ij} = \begin{cases} \frac{W_{ij}}{2||\mathbf{q}^i - \mathbf{q}^j||_2}, & |||\mathbf{q}^i - \mathbf{q}^j||_2 \le \varepsilon \\ 0, & otherwise \end{cases}$;
    2. Calculate $\widetilde{L}_t = \widetilde{D}_t - \widetilde{W}_t$, where $\widetilde{D}_t$ is a diagonal matrix with the $i$-th diagonal element as $\sum_j (\widetilde{W}_t)_{ij}$;
    3. Calculate $Q_{t+1} = (\widetilde{L}_t + I)^{-1} Y$;
    4. $t = t + 1$;
**end while**
**Output** The decision value matrix $Q_t \in \Re^{n\times c}$.
```

where $\varepsilon > 0$ is a small parameter and can be automatically selected, *e.g.* we can consider 5% of data as noise. In recent work [27, 28], the capped ℓ_1-norm was used to approximate the ℓ_0-norm with better results than ℓ_1-norm. When our new objective is minimized, the capped norm leads $||\mathbf{q}^i - \mathbf{q}^j||_2 = 0$ for many pairs (i, j) such that the ideal label assignments are achieved.

The other key challenge is to solve the above non-smooth and non-convex objective. All previous works used tedious algorithms to solve capped norm minimization problem. Based on our previous re-weighted optimization method [21, 22], we propose a novel, concise, and efficient algorithm to solve the proposed new objective.

Taking the derivative of Eq. (3) *w.r.t.* Q, and setting the derivative to zero, we have:

$$\widetilde{L}Q + (Q - Y) = \mathbf{0} \Rightarrow Q = (\widetilde{L} + I)^{-1} Y\,, \qquad (4)$$

where the new Laplacian matrix $\widetilde{L} = \widetilde{D} - \widetilde{W}$ and the re-weighted weight matrix $\widetilde{W}$ is defined by

$$\widetilde{W}_{ij} = \begin{cases} \frac{W_{ij}}{2||\mathbf{q}^i - \mathbf{q}^j||_2}, & |||\mathbf{q}^i - \mathbf{q}^j||_2 \le \varepsilon \\ 0, & otherwise \end{cases} \qquad (5)$$

$\widetilde{D}$ is a diagonal matrix with the i-th diagonal element as $\sum_j \widetilde{W}_{ij}$. Note that $(\widetilde{L} + I)^{-1}$ is dependent on Q, we propose an iterative algorithm to obtain the solution Q such that Eq. (4) is satisfied. The algorithm is guaranteed to converge to a local optimum, which will be proved in the next subsection. The algorithm is summarized in Algorithm 1.

4.3 Convergence Analysis

To prove the convergence of the Algorithm 1, we need the following lemma:

LEMMA 1. *For any nonzero vectors $\mathbf{q}, \mathbf{q}_t \in \Re^d$, the following inequality holds:*

$$||\mathbf{q}||_2 - \frac{||\mathbf{q}||_2^2}{2||\mathbf{q}_t||_2} \le ||\mathbf{q}_t||_2 - \frac{||\mathbf{q}_t||_2^2}{2||\mathbf{q}_t||_2}. \qquad (6)$$

PROOF. Beginning with an obvious inequality $-(||\mathbf{q}||_2 - ||\mathbf{q}_t||_2)^2 \le 0$, we have

$$\begin{aligned} &-(||\mathbf{q}||_2 - ||\mathbf{q}_t||_2)^2 \le 0 \Rightarrow 2||\mathbf{q}||_2||\mathbf{q}_t||_2 - ||\mathbf{q}||_2^2 \le ||\mathbf{q}_t||_2^2 \\ &\Rightarrow ||\mathbf{q}||_2 - \tfrac{||\mathbf{q}||_2^2}{2||\mathbf{q}_t||_2} \le ||\mathbf{q}_t||_2 - \tfrac{||\mathbf{q}_t||_2^2}{2||\mathbf{q}_t||_2} \end{aligned} \qquad (7)$$

which completes the proof. □

The following theorem guarantees that the Algorithm 1 will converge to the local optimum of the problem (3):

THEOREM 1. *The Algorithm 1 will monotonically decrease the objective of the problem (3) in each iteration, and converge to the local optimum of the problem.*

PROOF. According to the Step 3 in the algorithm 1, we know that:

$$\begin{aligned} Q_{t+1} = \arg\min_{Q} \sum_{i,j=1}^{n} \widetilde{W}_{ij} \min\{||\mathbf{q}^i - \mathbf{q}^j||_2, \varepsilon\} \\ +\gamma Tr(Q - Y)^T(Q - Y). \end{aligned} \qquad (8)$$

Because $\widetilde{W}_{ij} = \begin{cases} \frac{W_{ij}}{2||\mathbf{q}^i - \mathbf{q}^j||_2}, & ||\mathbf{q}^i - \mathbf{q}^j||_2 \le \varepsilon \\ 0, & otherwise \end{cases}$, we have:

$$\begin{aligned} &\sum_{k,l\in\mathcal{C}} \tfrac{W_{lk}||\mathbf{q}_{t+1}^l - \mathbf{q}_{t+1}^k||_2^2}{2||\mathbf{q}_t^l - \mathbf{q}_t^k||_2} + \gamma Tr(Q_{t+1} - Y)^T(Q_{t+1} - Y) \\ &\le \sum_{l,k\in\mathcal{C}} \tfrac{W_{lk}||\mathbf{q}_t^l - \mathbf{q}_t^k||_2^2}{2||\mathbf{q}_t^l - \mathbf{q}_t^k||_2} + \gamma Tr(Q_t - Y)^T(Q_t - Y)\,, \end{aligned} \qquad (9)$$

where $\mathcal{C}$ is the set in which $||\mathbf{q}^l - \mathbf{q}^k||_2 \le \varepsilon$ for all $k, l \in \mathcal{C}$.

Based on Lemma 1, we have:

$$||\mathbf{q}_{t+1}^l - \mathbf{q}_{t+1}^k||_2^2 - \frac{||\mathbf{q}_{t+1}^l - \mathbf{q}_{t+1}^k||_2^2}{2||\mathbf{q}_t^l - \mathbf{q}_t^k||_2} \le ||\mathbf{q}_t^l - \mathbf{q}_t^k||_2^2 - \frac{||\mathbf{q}_t^l - \mathbf{q}_t^k||_2^2}{2||\mathbf{q}_t^l - \mathbf{q}_t^k||_2}\,,$$

for any pair (l, k). Thus, for the set $\mathcal{C}$, we have:

$$\begin{aligned} &\sum_{k,l\in\mathcal{C}} W_{lk}(||\mathbf{q}_{t+1}^l - \mathbf{q}_{t+1}^k||_2^2 - \tfrac{||\mathbf{q}_{t+1}^l - \mathbf{q}_{t+1}^k||_2^2}{2||\mathbf{q}_t^l - \mathbf{q}_t^k||_2}) \\ &\le \sum_{k,l\in\mathcal{C}} W_{lk}(||\mathbf{q}_t^l - \mathbf{q}_t^k||_2^2 - \tfrac{||\mathbf{q}_t^l - \mathbf{q}_t^k||_2^2}{2||\mathbf{q}_t^l - \mathbf{q}_t^k||_2})\,. \end{aligned} \qquad (10)$$

Summing Eq. (9) and Eq. (10) on both sides, we obtain:

$$\begin{aligned} &\sum_{k,l\in\mathcal{C}} W_{ij}||\mathbf{q}_{t+1}^l - \mathbf{q}_{t+1}^k||_2 + \gamma Tr(Q_{t+1} - Y)^T(Q_{t+1} - Y) \\ &\le \sum_{k,l\in\mathcal{C}} W_{ij}||\mathbf{q}_t^l - \mathbf{q}_t^k||_2 + \gamma Tr(Q_t - Y)^T(Q_t - Y)\,, \end{aligned} \qquad (11)$$

which can be re-written as:

$$\begin{aligned} &\sum_{i,j=1}^{n} \widetilde{W}_{ij} \min\{||\mathbf{q}_{t+1}^i - \mathbf{q}_{t+1}^j||_2, \varepsilon\} + \gamma Tr(Q_{t+1} - Y)^T(Q_{t+1} - Y) \\ &\le \sum_{i,j=1}^{n} \widetilde{W}_{ij} \min\{||\mathbf{q}_t^i - \mathbf{q}_t^j||_2, \varepsilon\} + \gamma Tr(Q_t - Y)^T(Q_t - Y)\,. \end{aligned} \qquad (12)$$

Thus the Algorithm 1 will monotonically decrease the objective of the problem (3) in each iteration t. In the convergence, Q_t and $\widetilde{L}_t$ will satisfy the Eq. (4). As the problem (3) is a non-convex problem, satisfying the Eq. (4) indicates that Q_t is the a local optimum solution to the problem (3). Therefore, the Algorithm 3 will converge to the local optimum of the problem (3). □

5. PATTERN FUSION

In this step, we merge duplicated and complementary patterns to build templates in each cluster. Duplicates may caused by two reasons. One reason is that the same sentence generates duplicated tuples. For example, the two tuples {"they", "refused to divulge", "details"} and {"they", "refused", "to divulge details"} extracted from the same sentence. The other reason is different sentences generate the similar tuples. For example, {"a steady job", "working", "nights"} and {"he", "held", "a steady job working nights"} extracted from different sentences.

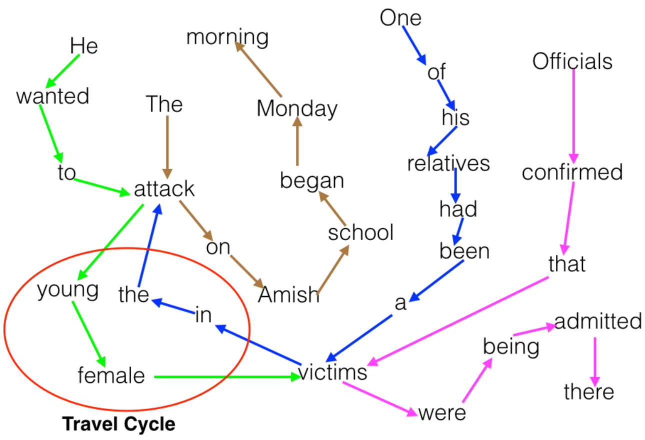

Figure 2: A word graph with cycle generated by tuple fusion

We follow graph-based pattern fusion approach proposed by Oya *et al.* [24]. Instead of using general pattern form of {arg1Type, predicate, arg2Type}, we use specific pattern form in which only replace head of arguments with it's type and keep other parts. Then, a graph is constructed by iteratively adding patterns to it. A node is added to the graph for each word in the pattern, and words adjacent are linked with directed edges. When adding a new pattern, a word from the pattern is merged to an existing node in the graph providing that they have the same POS tag and they share the same lemma. Note that some words such as "he" and "his" have the same POS tag "PRP" and the same lemma, but they should not be merged together. Also some words like "the", "to", "of" and "have" should not be merged together, unless it may lead to produce so many noisy paths. After word graph is constructed, we enumerate all paths from start node to end node. Figure 2 illustrates the word graph generated via tuple fusion instead of pattern fusion for convenience. We can see that the graph may contain cycles. Therefore, when the cycle appears, a backtracking procedure need to be executed.

6. SUMMARY GENERATION

In order to control wrong facts and generate meaningful sentences given pattern template, our system first filters out noisy templates by estimating the grammaticality using the recurrent neural network language model. Then, in order to generate meaningful sentences, we fuse tuples and select the best paths from tuple graph that satisfy two ranking criteria. One criterion is the path should best match the most fluent pattern template, and the other criterion is the path should be the most fluent path compared with other paths. Finally, we use Integer Linear programming to select top ranked facts cross clusters.

In order to filter out noisy templates, we leverage a neural network language model to rank the generated paths and select the top ranked one. We employ a character-based recurrent neural network language model [18] (RLM) interpolated with a Kneser-Ney trigram and find the n-best candidates with a stack decoder.

6.1 Recurrent Neural Network Language Model

A RLM models the probability $P(S)$ that the sequence of words S occurs in a given language. Let $S = w_1$, ... ,w_m be a sequence of m words:

$$P(S) = \prod_{i=1}^{m} P(w_i|w_{1:i-1}) \,. \tag{13}$$

The model explicitly computes without simplifying assumptions the conditional distributions $P(w_i|w_{1:i-1})$. The architecture of an RLM comprises a vocabulary V that contains the words w_i of the language as well as three transformations: an input vocabulary transformation $\mathbf{I} \in \mathcal{R}^{q*|V|}$, a recurrent transformation $\mathbf{R} \in \mathcal{R}^{q*q}$ and an output vocabulary transformation $\mathbf{O} \in \mathcal{R}^{|V|*q}$. For each word $w_k \in V$, we indicate by $i(w_k)$ its index in V and by $v(w_k) \in \mathcal{R}^{|V|*1}$ an all zero vector with only $v(w_k)_{i(w_k)} = 1$.

For a word w_i, the result of $\mathbf{I} \cdot v(w_i) \in \mathcal{R}^{q*1}$ is the input continuous representation of w_i. The parameter q governs the size of the word representation. The prediction proceeds by successively applying the recurrent transformation $\mathbf{R}$ to the word representations and predicting the next word at each step. In detail, the computation of each $P(w_i|w_{1:i-1})$ proceeds recursively. For $1 < i < m$,

$$h_1 = \sigma(\mathbf{I} \cdot v(w_1)) \tag{14}$$

$$h_{i+1} = \sigma(\mathbf{R} \cdot h_i + \mathbf{I} \cdot v(w_{i+1})) \tag{15}$$

$$o_{i+1} = \mathbf{O} \cdot h_i \,. \tag{16}$$

In above equations, σ is a nonlinear function such as $\tanh$. The conditional distribution is given by:

$$P(w_i = v|w_{1:i-1}) = \frac{\exp(o_{i,v})}{\sum_{v=1}^{V} \exp(o_{i,v})} \,. \tag{17}$$

The RLM is trained by back propagation through time [18]. The error in the predicted distribution calculated at the output layer is backpropagated through the recurrent layers and cumulatively added to the errors of the previous predictions for a given number d of steps.

6.2 Sentence Ranking

In order to find the best tuple path on tuple graph, we have devised the following ranking strategy. First, we prune the paths in which a verb does not exist. Then we rank other paths as follows:

Pattern Coverage

To identify the summary sentence with the highest coverage of select pattern, we propose a score to measure the similarity between pattern path and tuple path.

$$C(S) = \sum_{x_a \in Pa, x_b \in Tu} Sim(x_a, x_b) \tag{18}$$

Note that distributional semantics can be captured by continuous space word representation. We transform each token x into its embedding vector $\mathbf{x}$ by pre-trained distributed word representations, and then the similarity between a pair of x_a and x_b can be computed as their cosine similarity.

Fluency

In order to improve the grammaticality of the generated sentence, we estimate the grammaticality of generated paths $F(S)$ using the recurrent neural network language model.

In order to generate an abstractive sentence that combines the scores above, we employ a simple linear ranking model. The purpose of such a model is: 1) to cover the pattern optimally; 2) to generate a more readable and grammatical sentence. Therefore, the final ranking score of the path S is calculated over the normalized scores as:

$$Score(S) = \alpha \cdot C(S) + \beta \cdot F(S)\,, \tag{19}$$

where α and β are the coefficient factors to tune the ranking score and they sum up to 1. In order to rank the tuple graph paths, we select all the paths that contain at least one verb, and re-rank them using our proposed ranking function to find the best path as the summary of the original patterns in each cluster.

6.3 Sentence Selection

After sentence ranking, we prepare for the final abstractive summary generation. In this step, we select one sentence from each cluster. We use Integer Linear Programming to optimize a global objective function for sentence selection. We formulate the optimization problem based on sentence ranking information. More specifically, we would like to select exactly one sentence which receives the highest possible ranking score from each cluster subject to some constraints. We employed lp_solver [8], an efficient mixed integer programming solver using the Branch-and-Bound algorithm to select sentences.

Assume that there are in total K clusters in an event topic. For each cluster j, there are in total R ranked sentences. The variables S_{jl} is a binary indicator of the sentence. That is, S_{jl}= 1 if the sentence is included in the final summary, and S_{jl} = 0 otherwise. l is the ranked position of the sentence in this cluster.

Objective Function

Top ranked sentences are the most relevant corresponding to the related user interested aspects which we want to include in the final summary. Thus we try to minimize the ranks of the sentences to improve the overall responsiveness:

$$\min(\sum_{j=1}^{K}\sum_{l=1}^{R_j} l \cdot S_{jl})\,. \tag{20}$$

Exclusivity Constraints

To prevent redundancy in each aspect, we choose one sentence from each general or specific aspect cluster. The constraint is formulated as follows:

$$\sum_{l=1}^{R_j} S_{jl} = 1 \qquad \forall j \in \{1 \dots K\}\,. \tag{21}$$

Length Constraints

We add the following constraint to ensure that the length of the final summary is limited to L words. len_{jl} is the length of S_{jl}.

$$\sum_{j=1}^{K}\sum_{l=1}^{R_j} len_{jl} \cdot S_{jl} \leq L \tag{22}$$

7. EXPERIMENTAL RESULTS

7.1 Data Set Description

We use TAC2011 Summarization task [9] data set for the summary content evaluation. This data set provides 44 events. Each event falls into a predefined event topic. Each specific event includes an event statement and 20 relevant newswire articles which have been divided into 2 sets: Document Set A and Document Set B. Each document set has 10 documents, and all the documents in Set A chronologically precede the documents in Set B. We just use document Set A for our task. Assessors wrote model summaries for each event, so we can compare our automatic generated summaries with the model summaries.

7.2 Automatic Evaluation

We use the ROUGE [14] metric for measuring the summarization system performance. Ideally, a summarization criterion should be more recall oriented. So the average recall of ROUGE-1, ROUGE-2, ROUGE-SU4 were computed by running ROUGE-1.5.5 with stemming but no removal of stop words. We compare our method with the following baseline methods.

Baseline 1

In this baseline, we compare our method with traditional ranking and selection extractive summary generation framework [5] to show that the abstractive framework is superior to extractive framework in aspect-oriented summarization scenario. In implementation, we follow LexRank based sentence ranking combined with greedy sentence selection methods. The similarity graph building threshold is 0.2, damping factor is 0.2 and error tolerance for Power Method in LexRank is 0.1. After ranking sentences, we select top ranked sentences as long as the redundancy score (similarity) between a candidate sentence and current summary is under 0.5. This is repeated until the summary reaches a 100 word length limit.

Baseline 2

In this baseline, in order to prove the effectiveness of pattern fusion and surface realization, we replace our pattern fusion and surface realization with natural language generation technology. After grouping patterns via clustering, we choose the pattern that closest to centroid of the cluster. Then realize sentences with these representative patterns via natural language generation technology. More specifically, due to each pattern contains predicate, in order to generate sentence, we need generate noun phrase (See Algorithm 2) to build subject and generate verb phrase(See Algorithm 3) to build object. Then combine subject, predicate and object to become a new sentence. We use Simplenlg[10] which is a simple Java API designed to facilitate the realization of sentences.

Baseline 3

In this baseline, we try to compare our capped norm based semi-supervised learning model with the semi-supervised learning method using gaussian fields and harmonic functions (Har) [31] and the semi-supervised clustering with local and global consistency (LGC) [30].

To construct weight matrix, we use Heat Kernel (HK) [9] and Cosine similarity. Equation 23 illustrates how to construct weight matrix with Heat Kernel.

$$w_{i,j} = \exp(-\sum_{d=1}^{m} \frac{(x_{id} - x_{jd})^2}{\sigma_d^2})\,, \tag{23}$$

where x_{id} is the d-th component of pattern x_i represented as a continuous vector $x_i \in \mathcal{R}^m$ and $\sigma_1, \dots, \sigma_m$ are length scale hyper parameters for each dimension.

For fair comparison, all the baselines have the summary length no more then 100 words. In Table 3, we show the average ROUGE metrics of 44 summaries generated by our method and three baseline methods. Note that we present the best ROUGE scores after trying different σ settings.

[8] http://lpsolve.sourceforge.net/5.5/

[9] http://www.nist.gov/tac/2011/Summarization/

[10] http://code.google.com/p/simplenlg/

Algorithm 2 Generate Noun Phrase

```
$NPPhraseSpec\ np, tmpNp$;
$PPPhraseSpec\ pp$;
$Stack\ stack$
$stack.add(head)$
while $!stack.isEmpty()$ do
    $children = grpah.adj(head)$;
    for all $td$ in $children$ do
        $GrammaticalRelation\ gr = td.reln()$
        if $gr =$ "$prep$" then
            $pp = generatePrepP(td)$
            $np.setPostModifier(pp)$
        else if $gr =$ "$nn$" or $gr =$ "$conj$" then
            $tmpNp = generateNP(graph, td.dep())$
            $np.setPostModifier(tmpNp)$
        else if $gr =$ "$det$" or $gr =$ "$num$" or $gr =$ "$amod$" then
            $np.setPostModifier(td.dep())$
        else
            continue
        end if
    end for
end while
```

Results and Discussions

We compared our capped norm based semi-supervised learning model to two other popularly used semi-supervised learning methods: Harmonic function (Har) [31] and LGC [30]. In the comparison results, our new model consistently outperforms two other methods. Because we utilize the capped norm based loss, our new objective can achieve better semi-supervised learning label indicator matrix than traditional methods which use the ℓ_2-norm based loss function.

Our abstractive aspect-oriented summarization system shows statistically significant improvements over Baselines 2 and 3 and pure extractive summarization systems for ROUGE. This means our systems can effectively aggregate the extracted patterns and generate abstract sentences based on the relevant keywords. We can also observe that our abstractive summarization system produces the highest ROUGE-1 score among all models, which further confirms the success of our framework that can cover more human-authored words. However, compared with TAC 2011 best system and compression based approaches [12] that use extractive based supervised learning, our system performance is bad. This proved that fully unsupervised abstractive summarization is a very challenge task, however our pattern based approach shows the feasibility and usefulness of this new direction (our framework only uses a small number of hand-crafted keywords for aspect-oriented summarization and all the rest modules are unsupervised).

7.3 Manual Evaluation

To judge the quality of generated summaries, we ask three graduate students to score them. The judges will give a grammaticality and coherence scores to each summary. These two scores reflect the fluency and readability of the summary. Also the judges will give a informativeness score to each summary. This score reflects the coverage of all required aspects. The judges follow 5-point Likert scale to score each summary. We then compute the average scores. Note that we use LGC in Baselines 2 and 3 because they can get relative better clustering results. The manual evaluation results are shown in Table 4.

Algorithm 3 Generate Verb Phrase

```
$VPPhraseSpec\ vp$
$NPPhraseSpec\ dirobjNp, indirObjNp$
$vp.sertVerb(verb)$
if $object! = null$ then
    $dirobjNp = generateNP(graph, object)$
    $vp.setObject(dirobjNp)$
    $children = grpah.adj(verb)$;
    for all $td$ in $children$ do
        $GrammaticalRelation\ gr = td.reln()$
        if $gr =$ "$iobj$" then
            $indirObjNp = generateNP(graph, td.dep())$
            $vp.IndirectObject(indirObjNp)$
            break
        end if
    end for
else
    for all $td$ in $children$ do
        $GrammaticalRelation\ gr = td.reln()$
        if $gr =$ "$ccomp$" then
            $vp.setPostModifier(complement)$
            break
        end if
    end for
end if
```

Methods		Rouge Average Recall		
		R-1	R-2	R-SU4
BL-1	Lexrank	28.93	5.17	9.15
BL-2	Har+HK+NLG	29.64	4.14	8.53
	Har+Cosine+NLG	29.33	4.3	8.59
	LGC+HK+NLG	29.23	4.58	8.68
	LGC+Cosine+NLG	29.5	4.38	8.52
	Capped+HK+NLG	27.58	4.12	8.11
	Capped+Cosine+NLG	28.04	4.26	8.21
BL-3	Har+HK+Fuse	30.54	5.03	9.18
	Har+Cosine+Fuse	30.7	5.42	9.27
	LGC+HK+Fuse	30.6	5.04	9.26
	LGC+Cosine+Fuse	30.94	5.37	9.48
Ours	Capped+HK + Fuse	31.72	5.75	10.03
	Capped+Cosine + Fuse	**31.88**	**5.98**	**10.27**
TAC'11 best	Supervised extractive	n/a	13.44	16.51
Li et al., 2014	Supervised compression	n/a	14.4	16.89

Table 3: ROUGE evaluation results on TAC2011 Summarization data sets, The improvements made by our method over the baselines are all statistically significant at 95% confidence level (p<0.05).

Results and Discussions

As expected, Baseline 1 received the highest rating of grammaticality because it use extractive based approach. Baseline 2 received the lowest rating of grammaticality, this result reflect heuristic summary generation via construct subject and object still need more linguistic knowledge. Referring to coherence rating, Baseline 1 is the lowest due to other methods leverage hand crafted keywords as hints to improve final summary coherence. Referring to informativeness rating, our label propagation clustering can better find semantic similar and complementary patterns that match user interested aspects. Overall, abstractive summary generation via label propagation, tuple/pattern fusion and recurrent neural network language model can improve the fluency, readability and coverage of final summary.

Method		Human judegs Average Score		
		Grammaticality	Coherence	Informativeness
BL-1	Ext	3.4	2.7	3.2
BL-2	NLG	2.9	3.0	3.3
BL-3	Fuse	3.1	3.2	3.4
Our Method		3.3	3.3	3.6

Table 4: Manual evaluation results on TAC2011 Summarization data sets

7.4 Example of Output Summaries

Table 5 presents the output summary of the subject "Amish Shooting" and " China Water Shortage" generated by our method. Comparing against the human-authored summary, our method can capture additional information related to damage of the accident such as "Six people were dead." On the other hand, we realized that n-gram cooccurance based ROUGE metric may not be suitable for abstractive summarization, and the new sentences generated by our approach may not match human-authored golden standard summary well.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we study the problem of using pattern based approaches to generate abstractive summarization. Our system first generates meaningful patterns from sentences. In order to precisely cluster patterns, we propose a new capped norm based semi-supervised pattern learning algorithm that leverages a hand-crafted list of topic-relevant keywords. Our system generates new patterns by fusing existing patterns and selects top ranked new patterns via the recurrent neural network language model. Finally, we use Integer Linear programming to select top ranked facts cross clusters. Although fully unsupervised abstractive summarization is a challenging task (the performance is still low compared with supervised extractive summarization system), our work shows the feasibility and usefulness of this new direction for summarization research.

There are a number of directions we plan to pursue in the future in order to improve our method. First, we can possibly apply more linguistic knowledge to improve the quality of head type tagging. Currently the tagging pipeline may generate meaningless tags. Second, we may explore more domain knowledge to improve the quality of pattern clustering. For example, we know that the "who-affected" aspect is related to person, and "when, where" are related to time and location, we can leverage these annotated phrases to help group relevant sentences. Third, we want to enhance our semi-supervised clustering model to precisely find similar patterns.

9. ACKNOWLEDGMENTS

This work was supported in part by Australian Research Council (ARC) grants and U.S. NSF-IIS 1117965, NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628.

Human-Authored Summary: D1101A-A
On October 2, 2006, a gunman, Charles Roberts, entered an Amish school near Lancaster, PA, took the children hostage, killed five girls and wounded seven other children before killing himself. His wife said that Roberts, a truck driver from a nearby town, told her he was acting in revenge for something that happened 20 years ago. He believed that he had molested two relatives. They denied that this ever happened. He also had anguish over his infant daughter's death. Roberts' wife said he was an exceptional father and a loving husband. The Amish community reached out to Roberts' family.

Summary generated by our system: D1101A-A
They were not molested 20 years ago. Two state troopers on horseback and bonnets stood two hours after the shooting. A procession of 34 buggies and graveyard was deserted Monday. Six people were dead. Charles Carl Roberts IV held a steady job working nights driving a truck that collected milk from area dairy farms. State police have said Roberts called her husband an exceptional father who took his children to soccer practice. These families to be left alone.

Human-Authored Summary: D11036G-A
On 3 January, 2007, a report from Beijing stated that global warming is seriously impacting China's ecological, social, and economic systems, and some of this damage will be irreversible. And while concerned about global warming, China says it lacks the money/technology to reduce its greenhouse gas emissions. The most serious of these issues is water shortages. The Yangtze River, the nation's largest, is at its lowest level in 100 years. This shortage is affecting fish and migratory bird survivals, crop productions, availability of drinking water, and numerous maritime shipping problems. China does plan to cut water consumption 20% by 2010.

Summary generated by our system: D1136G-A
Feb. 22, East China is among the driest country as water evaporates more rapidly from river are connected the Yangtze threads its way through 11 regions in Beijing. The floodgates could release an additional 6.1 billion cubic meters per capita income of over 50,000 fishermen around Poyang Lake was less than 600 yuan. The city had 15 million permanent residents and neighboring Sichuan Province in China plans to cut its industrial water per day, only 35 centimeters higher than the record low. Yangtze fishermen are experiencing a record low while its tributaries dropped 30 to 60 percent below average levels

Table 5: A comparison between a human-authored summary and a summary generated by our system

10. REFERENCES

[1] E. Alfonseca, D. Pighin, and G. Garrido. Heady: News headline abstraction through event pattern clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1243–1253, Sofia, Bulgaria, 2013.

[2] N. Balasubramanian, S. Soderland, Mausam, and O. Etzioni. Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1721–1731, Seattle, Washington, USA, 2013.

[3] Y.-N. Chen, W. Y. Wang, and A. I. Rudnicky. Leveraging frame semantics and distributional semantics for unsupervised slot induction for spoken dialogue systems. In

Extened Abstract Presented at The 52nd Annual Meeting of Association for Computational Linguistics 2014 Workshop on Semantic Parsing (ACL-SP 2014), Baltimore, MD, USA, 2014. ACL.

[4] J. M. Conroy, J. D. Schlesinger, P. A. Rankel, and D. P. ÒLeary. Classy 2010: Summarization and metrics. In *Proceedings of the Third Text Analysis Conference, Gaithersburg, Maryland, USA. National Institute of Standards and Technology*, 2010.

[5] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 2004.

[6] P.-E. Genest and G. Lapalme. Text generation for abstractive summarization. In *Proceedings of the Third Text Analysis Conference, Gaithersburg, Maryland, USA. National Institute of Standards and Technology*, 2010.

[7] P.-E. Genest and G. Lapalme. Fully abstractive approach to guided summarization. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 354–358, Stroudsburg, PA, USA, 2012.

[8] D. Gillick, B. Favre, D. Hakkani-Tur, B. Bohnet, Y. Liu, and S. Xie. The icsi/utd summarization system at tac 2009. In *The ICSI/UTD Summarization System at TAC 2009*, 2009.

[9] X. He and P. Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, volume 16, page 153, 2004.

[10] W. Jiang, F. Nie, and H. Huang. Robust dictionary learning with capped l1 norm. *Twenty-Fourth International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 3590–3596, 2015.

[11] K. S. Jones. Automatic summarizing: factors and directions. *Advances in automatic text summarization*, pages 1–12, 1999.

[12] C. Li, Y. Liu, F. Liu, L. Zhao, and F. Weng. Improving multi-documents summarization by sentence compression based on expanded constituent parse trees. *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[13] P. Li, Y. Wang, W. Gao, and J. Jiang. Generating aspect-oriented multi-document summarization with event-aspect model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 2011.

[14] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *NAACL*, 2003.

[15] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 523–534, Stroudsburg, PA, USA, 2012.

[16] Y. Mehdad, G. Carenini, and R. T. Ng. Abstractive summarization of spoken and written conversations based on phrasal queries. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1220–1230, Baltimore, Maryland, 2014.

[17] Y. Mehdad, G. Carenini, F. W. Tompa, and R. Ng. Abstractive meeting summarization with entailment and fusion. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 136–146, 2013.

[18] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048, 2010.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[20] G. Murray, G. Carenini, and R. Ng. Generating and validating abstracts of meeting conversations: a user study. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 105–113, 2010.

[21] F. Nie, H. Huang, X. Cai, and C. Ding. Efficient and robust feature selection via joint l2,1-norms minimization. *Neural Information Processing Systems Conference (NIPS)*, pages 1813–1821, 2010.

[22] F. Nie, H. Huang, C. Ding, D. Luo, and H. Wang. Principal component analysis with non-greedy l1-norm maximization. *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1433–1438, 2011.

[23] F. Nie, H. Wang, H. Huang, and C. Ding. Unsupervised and semi-supervised learning via l1-norm graph. *IEEE Conference on Computer Vision (ICCV)*, pages 2268–2273, 2011.

[24] T. Oya, Y. Mehdad, G. Carenini, and R. Ng. A template-based abstractive meeting summarization: Leveraging summary and source text relationships. In *Proceedings of the 8th International Natural Language Generation Conference (INLG)*, pages 45–53, Philadelphia, Pennsylvania, U.S.A., 2014.

[25] D. Pighin, M. Cornolti, E. Alfonseca, and K. Filippova. Modelling events through memory-based, open-ie patterns for abstractive summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–901, Baltimore, Maryland, 2014.

[26] L. Wang and C. Cardie. Domain-independent abstract generation for focused meeting summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1395–1405, Sofia, Bulgaria, 2013.

[27] T. Zhang. Multi-stage convex relaxation for learning with sparse regularization. *NIPS*, 2008.

[28] T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *JMLR*, pages 1081–1107, 2010.

[29] X. Zhang and M. Lapata. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 670–680, 2014.

[30] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16(16):321–328, 2004.

[31] X. Zhu, Z. Ghahramani, J. Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919, 2003.