Domain-Independent Abstract Generation for Focused Meeting Summarization

Lu Wang
Department of Computer Science
Cornell University
Ithaca, NY 14853
luwang@cs.cornell.edu

Claire Cardie
Department of Computer Science
Cornell University
Ithaca, NY 14853
cardie@cs.cornell.edu

Abstract

We address the challenge of generating natural language abstractive summaries for spoken meetings in a domain-independent fashion. We apply *Multiple-Sequence Alignment* to induce abstract generation templates that can be used for different domains. An *Overgenerate-and-Rank* strategy is utilized to produce and rank candidate abstracts. Experiments using in-domain and out-of-domain training on disparate corpora show that our system uniformly outperforms state-of-the-art supervised extract-based approaches. In addition, human judges rate our system summaries significantly higher than compared systems in fluency and overall quality.

1 Introduction

Meetings are a common way to collaborate, share information and exchange opinions. Consequently, automatically generated meeting summaries could be of great value to people and businesses alike by providing quick access to the essential content of past meetings. *Focused meeting summaries* have been proposed as particularly useful; in contrast to summaries of a meeting as a whole, they refer to summaries of a specific aspect of a meeting, such as the DECISIONS reached, PROBLEMS discussed, PROGRESS made or ACTION ITEMS that emerged (Carenini et al., 2011). Our goal is to provide an automatic summarization system that can generate abstract-style focused meeting summaries to help users digest the vast amount of meeting content in an easy manner.

Existing meeting summarization systems remain largely *extractive*: their summaries are comprised exclusively of patchworks of utterances selected directly from the meetings to be summarized (Riedhammer et al., 2010; Bui et al., 2009; Xie et al., 2008). Although relatively easy to construct, extractive approaches fall short of producing concise and readable summaries, largely due to the noisy, fragmented, ungrammatical and unstructured text of meeting transcripts (Murray et al., 2010b; Liu and Liu, 2009).

C: Looking at what we've got, we we want an LCD display with a spinning wheel.
B: You have to have some push-buttons, don't you?
C: Just spinning and not scrolling, I would say.
B: I think the spinning wheel is definitely very now.
A: but since LCDs seems to be uh a definite yes,
C: We're having push-buttons on the outside
C: and then on the inside an LCD with spinning wheel,

Decision Abstract (Summary):
The remote will have push buttons outside, and an LCD and spinning wheel inside.

A: and um I'm not sure about the buttons being in the shape of fruit though.
D: Maybe make it like fruity colours or something.
C: The power button could be like a big apple or something.
D: Um like I'm just thinking bright colours.

Problem Abstract (Summary):
How to incorporate a fruit and vegetable theme into the remote.

Figure 1: Clips from the AMI meeting corpus (Mccowan et al., 2005). A, B, C and D refer to distinct speakers. Also shown is the gold-standard (manual) abstract (summary) for the decision and the problem.

In contrast, human-written meeting summaries are typically in the form of *abstracts* — distillations of the original conversation written in new language. A user study from Murray et al. (2010b) showed that people demonstrate a strong preference for abstractive summaries over extracts when the text to be summarized is conversational. Consider, for example, the two types of focused summary along with their associated dialogue snippets in Figure 1. We can see that extracts are likely to include unnecessary and noisy information from the meeting transcripts. On the contrary, the manually composed summaries (abstracts) are more compact and readable, and are written in a distinctly non-conversational style.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1395–1405,
Sofia, Bulgaria, August 4-9 2013.

To address the limitations of extract-based summaries, we propose a complete and fully automatic domain-independent abstract generation framework for focused meeting summarization. Following existing language generation research (Angeli et al., 2010; Konstas and Lapata, 2012), we first perform *content selection*: given the dialogue acts relevant to one element of the meeting (e.g. a single decision or problem), we train a classifier to identify summary-worthy phrases. Next, we develop an "overgenerate-and-rank" strategy (Walker et al., 2001; Heilman and Smith, 2010) for *surface realization*, which generates and ranks candidate sentences for the abstract. After redundancy reduction, the full meeting abstract can thus comprise the focused summary for each meeting element. As described in subsequent sections, the generation framework allows us to identify and reformulate the important information for the focused summary. Our contributions are as follows:

- To the best of our knowledge, our system is the first fully automatic system to generate natural language abstracts for spoken meetings.

- We present a novel template extraction algorithm, based on Multiple Sequence Alignment (MSA) (Durbin et al., 1998), to induce domain-independent templates that guide abstract generation. MSA is commonly used in bioinformatics to identify equivalent fragments of DNAs (Durbin et al., 1998) and has also been employed for learning paraphrases (Barzilay and Lee, 2003).

- Although our framework requires labeled training data for each type of focused summary (decisions, problems, etc.), we also make initial tries for domain adaptation so that our summarization method does not need human-written abstracts for each new meeting domain (e.g. faculty meetings, theater group meetings, project group meetings).

We instantiate the abstract generation framework on two corpora from disparate domains — the AMI Meeting Corpus (Mccowan et al., 2005) and ICSI Meeting Corpus (Janin et al., 2003) — and produce systems to generate focused summaries with regard to four types of meeting elements: DECISIONs, PROBLEMs, ACTION ITEMSs, and PROGRESS. Automatic evaluation (using ROUGE (Lin and Hovy, 2003) and BLEU (Papineni et al., 2002)) against manually generated focused summaries shows that our summarizers uniformly and statistically significantly outperform two baseline systems as well as a state-of-the-art supervised extraction-based system. Human evaluation also indicates that the abstractive summaries produced by our systems are more linguistically appealing than those of the utterance-level extraction-based system, preferring them over summaries from the extraction-based system of comparable semantic correctness (62.3% vs. 37.7%).

Finally, we examine the generality of our model across domains for two types of focused summarization — decisions and problems — by training the summarizer on out-of-domain data (i.e. the AMI corpus for use on the ICSI meeting data, and vice versa). The resulting systems yield results comparable to those from the same system trained on in-domain data, and statistically significantly outperform supervised extractive summarization approaches trained on in-domain data.

2 Related Work

Most research on spoken dialogue summarization attempts to generate summaries for full dialogues (Carenini et al., 2011). Only recently has the task of focused summarization been studied. Supervised methods are investigated to identify key phrases or utterances for inclusion in the decision summary (Fernández et al., 2008; Bui et al., 2009). Based on Fernández et al. (2008), a relation representation is proposed by Wang and Cardie (2012) to form structured summaries; we adopt this representation here for content selection.

Our research is also in line with generating abstractive summaries for conversations. Extractive approaches (Murray et al., 2005; Xie et al., 2008; Galley, 2006) have been investigated extensively in conversation summarization. Murray et al. (2010a) present an abstraction system consisting of interpretation and transformation steps. Utterances are mapped to a simple conversation ontology in the interpretation step according to their type, such as a decision or problem. Then an integer linear programming approach is employed to select the utterances that cover more entities as

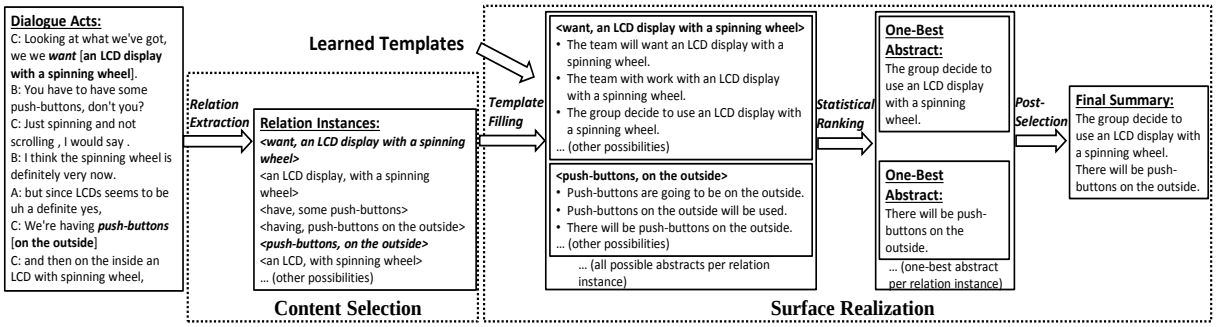

Figure 2: The abstract generation framework. It takes as input a cluster of meeting-item-specific dialogue acts, from which one focused summary is constructed. Sample relation instances are denoted in **bold** (The indicators are further *italicized* and the arguments are in [brackets]). Summary-worthy relation instances are identified by content selection module (see Section 4) and then filled into the learned templates individually. A statistical ranker subsequently selects one best abstract per relation instance (see Section 5.2). The post-selection component reduces the redundancy and outputs the final summary (see Section 5.3).

determined by an external ontology. Liu and Liu (2009) apply sentence compression on extracted summary utterances. Though some of the unnecessary words are dropped, the resulting compressions can still be ungrammatical and unstructured.

This work is also broadly related to expert system-based language generation (Reiter and Dale, 2000) and concept-to-text generation tasks (Angeli et al., 2010; Konstas and Lapata, 2012), where the generation process is decomposed into content selection (or text planning) and surface realization. For instance, Angeli et al. (2010) learn from structured database records and parallel textual descriptions. They generate texts based on a series of decisions made to select the records, fields, and proper templates for rendering. Those techniques that are tailored to specific domains (e.g. weather forecasts or sportcastings) cannot be directly applied to the conversational data, as their input is well-structured and the templates learned are domain-specific.

3 Framework

Our domain-independent abstract generation framework produces a summarizer that generates a grammatical abstract from a cluster of *meeting-element-related dialogue acts (DAs)* — all utterances associated with a single decision, problem, action item or progress step of interest. Note that identifying these DA clusters is a difficult task in itself (Bui et al., 2009). Accordingly, our experiments evaluate two conditions — one in which we assume that they are perfectly identified, and one in which we identify the clusters automatically.

The summarizer consists of two major components and is depicted in Figure 2. Given the DA cluster to be summarized, the *Content Selection* module identifies a set of summary-worthy *relation instances* represented as indicator-argument pairs (i.e. these constitute a finer-grained representation than DAs). The *Surface Realization* component then generates a short summary in three steps. In the first step, each relation instance is filled into templates with disparate structures that are learned automatically from the training set (*Template Filling*). A statistical ranker then selects one best abstract per relation instance (*Statistical Ranking*). Finally, selected abstracts are processed for redundancy removal in *Post-Selection*. Detailed descriptions for each individual step are provided in Sections 4 and 5.

4 Content Selection

Phrase-based content selection approaches have been shown to support better meeting summaries (Fernández et al., 2008). Therefore, we chose a content selection representation of a finer granularity than an utterance: we identify *relation instances* that can both effectively detect the crucial content and incorporate enough syntactic information to facilitate the downstream surface realization.

More specifically, our relation instances are based on information extraction methods that identify a lexical *indicator* (or *trigger*) that evokes a relation of interest and then employ syntactic information, often in conjunction with semantic constraints, to find the *argument constituent*(or *target phrase*) to be extracted. Rela-

tion instances, then, are represented by **indicator-argument** pairs (Chen et al., 2011). For example, in the DA cluster of Figure 2, ⟨*want, an LCD display with a spinning wheel*⟩ and ⟨*push-buttons, on the outside*⟩ are two relation instances.

Relation Instance Extraction We adopt and extend the syntactic constraints from Wang and Cardie (2012) to identify all relation instances in the input utterances; the summary-worthy ones will be selected by a discriminative classifier. Constituent and dependency parses are obtained by the Stanford parser (Klein and Manning, 2003). Both the indicator and argument take the form of constituents in the parse tree. We restrict the eligible indicator to be a noun or verb; the eligible arguments is a noun phrase (NP), prepositional phrase (PP) or adjectival phrase (ADJP). A valid indicator-argument pair should have at least one content word and satisfy one of the following constraints:

- When the indicator is a noun, the argument has to be a modifier or complement of the indicator.
- When the indicator is a verb, the argument has to be the subject or the object if it is an NP, or a modifier or complement of the indicator if it is a PP/ADJP.

We view relation extraction as a binary classification problem rather than a clustering task (Chen et al., 2011). All relation instances can be categorized as summary-worthy or not, but only the summary-worthy ones are used for abstract generation. A discriminative classifier is trained for this purpose based on Support Vector Machines (SVMs) (Joachims, 1998) with an RBF kernel. For training data construction, we consider a relation instance to be a positive example if it shares any content word with its corresponding abstracts, and a negative example otherwise. The features used are shown in Table 1.

5 Surface Realization

In this section, we describe surface realization, which renders the relation instances into natural language abstracts. This process begins with template extraction (Section 5.1). Once the templates are learned, the relation instances from Section 4 are filled into the templates to generate an abstract (see Section 5.2). Redundancy handling is discussed in Section 5.3.

Basic Features
number of words/content words
portion of content words/stopwords
number of content words in indicator/argument
number of content words that are also in previous DA
indicator/argument only contains stopword?
number of new nouns
Content Features
has capitalized word?
has proper noun?
TF/IDF/TFIDF min/max/average
Discourse Features
main speaker or not?
is in an adjacency pair (AP)?
is in the source/target of the AP?
number of source/target DA in the AP
is the target of the AP a positive/negative/neutral response?
is the source of the AP a question?
Syntax Features
indicator/argument constituent tag
dependency relation of indicator and argument

Table 1: Features for content selection. Most are adapted from previous work (Galley, 2006; Xie et al., 2008; Wang and Cardie, 2012). Every basic or content feature is concatenated with the constituent tags of indicator and argument to compose a new one. Main speakers include the most talkative speaker (who has said the most words) and other speakers whose word count is more than 20% of the most talkative one (Xie et al., 2008). Adjacency pair (AP) (Galley, 2006) is an important conversational analysis concept; each AP consists of a source utterance and a target utterance produced by different speakers.

5.1 Template Extraction

Sentence Clustering. Template extraction starts with clustering the sentences that constitute the manually generated abstracts in the training data according to their lexical and structural similarity. From each cluster, multiple-sequence alignment techniques are employed to capture the recurring patterns.

Intuitively, desirable templates are those that can be applied in different domains to generate the same type of focused summary (e.g. decision or problem summaries). We do not want sentences to be clustered only because they describe the same domain-specific details (e.g. they are all about "data collection"), which will lead to fragmented templates that are not reusable for new domains. We therefore replace all appearances of dates, numbers, and proper names with generic labels. We also replace words that appear in both the abstract and supporting dialogue acts by a label indicating its phrase type. For any noun phrase with its head word abstracted, the whole phrase is also replaced with "NP".

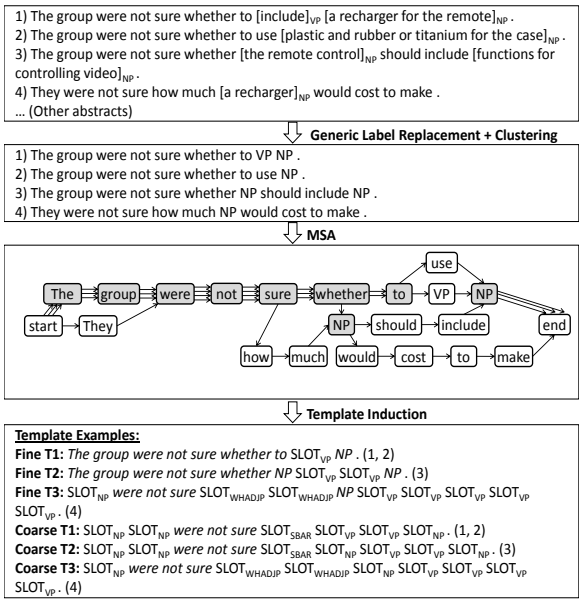

Figure 3: Example of template extraction by Multiple-Sequence Alignment for problem abstracts from AMI. Backbone nodes shared by at least 50% sentences are shaded. The grammatical errors exist in the original abstracts.

Following Barzilay and Lee (2003), we approach the sentence clustering task by hierarchical complete-link clustering with a similarity metric based on word n-gram overlap ($n = 1, 2, 3$). Clusters with fewer than three abstracts are removed[1].

Learning the Templates via MSA. For learning the structural patterns among the abstracts, *Multiple-Sequence Alignment (MSA)* is first computed for each cluster. MSA takes as input multiple sentences and one scoring function to measure the similarity between any two words. For insertions or deletions, a gap cost is also added. MSA can thus find the best way to align the sequences with insertions or deletions in accordance with the scorer. However, computing an optimal MSA is NP-complete (Wang and Jiang, 1994), thus we implement an approximate algorithm (Needleman and Wunsch, 1970) that iteratively aligns two sequences each time and treats the resulting alignment as a new sequence[2]. Figure 3 demonstrates an MSA computed from a sample cluster of abstracts. The MSA is represented in the form of word lattice, from which we can detect the structural similarities shared by the sentences.

To transform the resulting MSAs into templates, we need to decide whether a word in the sentence should be retained to comprise the template or abstracted. The *backbone* nodes in an MSA are identified as the ones shared by more than 50%[3] of the cluster's sentences (shaded in gray in Figure 3). We then create a FINE template for each sentence by abstracting the non-backbone words, i.e. replacing each of those words with a generic token (last step in Figure 3). We also create a COARSE template that only preserves the nodes shared by all of the cluster's sentences. By using the operations above, domain-independent patterns are thus identified and domain-specific details are removed.

Note that we do not explicitly evaluate the quality of the learned templates, which would require a significant amount of manual evaluation. Instead, they are evaluated extrinsically. We encode the templates as features (Angeli et al., 2010) that could be selected or ignored in the succeeding abstract ranking model.

5.2 Template Filling

An Overgenerate-and-Rank Approach. Since filling the relation instances into templates of distinct structures may result in abstracts of varying quality, we rank the abstracts based on the features of the template, the transformation conducted, and the generated abstract. This is realized by the *Overgenerate-and-Rank* strategy (Walker et al., 2001; Heilman and Smith, 2010). It takes as input a set of relation instances (from the same cluster) $R = \{\langle ind_i, arg_i \rangle\}_{i=1}^{N}$ that are produced by content selection component, a set of templates $T = \{t_j\}_{j=1}^{M}$ that are represented as parsing trees, a transformation function F (described below), and a statistical ranker S for ranking the generated abstracts, for which we defer description later in this Section.

For each $\langle ind_i, arg_i \rangle$, the overgenerate-and-rank approach fills it into each template in T by applying F to generate all possible abstracts. Then the ranker S selects the best abstract abs_i. Post-selection is conducted on the abstracts $\{abs_i\}_{i=1}^{N}$ to form the final summary.

[1]Clustering stops when the similarity between any pairwise clusters is below 5. This is applied to every type of summarization. We tune the parameter on a small held-out development set by manually evaluating the induced templates. No significant change is observed within a small range.

[2]We adopt the scoring function for MSA from Barzilay and Lee (2003), where aligning two identical words scores 1, inserting a gap scores -0.01, and aligning two different words scores -0.5.

[3]See Barzilay and Lee (2003) for a detailed discussion about the choice of 50% according to pigeonhole principle.

The transformation function F models the *constituent-level* transformations of relation instances and their mappings to the parse trees of templates. With the intuition that people will reuse the relation instances from the transcripts albeit not necessarily in their original form to write the abstracts, we consider three major types of mapping operations for the indicator or argument in the source pair, namely, *Full-Constituent Mapping*, *Sub-Constituent Mapping*, and *Removal*. *Full-Constituent Mapping* denotes that a source constituent is mapped directly to a target constituent of the template parse tree with the same tag. *Sub-Constituent Mapping* encodes more complex and flexible transformations in that a sub-constituent of the source is mapped to a target constituent with the same tag. This operation applies when the source has a tag of PP or ADJP, in which case its sub-constituent, if any, with a tag of NP, VP or ADJP can be mapped to the target constituent with the same tag. For instance, an argument "with a spinning wheel" (PP) can be mapped to an NP in a template because it has a sub-constituent "a spinning wheel" (NP). *Removal* means a source is not mapped to any constituent in the template.

Formally, F is defined as:

$$F(\langle ind^{src}, arg^{src}\rangle, t) = \{\langle ind_k^{tran}, arg_k^{tran}, ind_k^{tar}, arg_k^{tar}\rangle\}_{k=1}^{K}$$

where $\langle ind^{src}, arg^{src}\rangle \in R$ is a relation instance (*source pair*); $t \in T$ is a template; ind_k^{tran} and arg_k^{tran} is the *transformed pair* of ind^{src} and arg^{src}; ind_k^{tar} and arg_k^{tar} are constituents in t, and they compose one *target pair* for $\langle ind^{src}, arg^{src}\rangle$. We require that ind^{src} and arg^{src} are not removed at the same time. Moreover, for valid ind_k^{tar} and arg_k^{tar}, the words subsumed by them should be all abstracted in the template, and they do not overlap in the parse tree.

To obtain the realized abstract, we traverse the parse tree of the filled template in pre-order. The words subsumed by the leaf nodes are thus collected sequentially.

Learning a Statistical Ranker. We utilize a discriminative ranker based on Support Vector Regression (SVR) (Smola and Schölkopf, 2004) to rank the generated abstracts. Given the training data that includes clusters of gold-standard summary-worthy relation instances, associated abstracts they support, and the parallel templates for each abstract, training samples for the ranker are constructed according to the transformation function F mentioned above. Each sample is represented as:

$(\langle ind^{src}, arg^{src}\rangle, \langle ind_k^{tran}, arg_k^{tran}, ind_k^{tar}, arg_k^{tar}\rangle, t, a)$

where $\langle ind^{src}, arg^{src}\rangle$ is the source pair, $\langle ind_k^{tran}, arg_k^{tran}\rangle$ is the transformed pair, $\langle ind_k^{tar}, arg_k^{tar}\rangle$ is the target pair in template t, and a is the abstract parallel to t.

We first find $\langle ind_k^{tar,abs}, arg_k^{tar,abs}\rangle$, which is the corresponding constituent pair of $\langle ind_k^{tar}, arg_k^{tar}\rangle$ in a. Then we identify the summary-worthy words subsumed by $\langle ind_k^{tran}, arg_k^{tran}\rangle$ that also appear in a. If those words are all subsumed by $\langle ind_k^{tar,abs}, arg_k^{tar,abs}\rangle$, then it is considered to be a positive sample, and a negative sample otherwise. Table 2 displays the features used in abstract ranking.

Basic Features
number of words in ind^{src}/arg^{src}
number of new nouns in ind^{src}/arg^{src}
ind_k^{tran}/arg_k^{tran} only has stopword?
number of new nouns in ind_k^{tran}/arg_k^{tran}
Structure Features
constituent tag of ind^{src}/arg^{src}
constituent tag of ind^{src} with constituent tag of ind^{tar}
constituent tag of arg^{src} with constituent tag of arg^{tar}
transformation of ind^{src}/arg^{src} combined with constituent tag
dependency relation of ind^{src} and arg^{src}
dependency relation of ind^{tar} and arg^{tar}
above 2 features have same value?
Template Features
template type (fine/coarse)
realized template (e.g. "the group decided to")
number of words in template
the template has verb?
Realization Features
realization has verb?
realization starts with verb?
realization has adjacent verbs/NPs?
ind^{src} precedes/succeeds arg^{src}?
ind^{tar} precedes/succeeds arg^{tar}?
above 2 features have same value?
Language Model Features
$\log p_{LM}$(first word in ind_k^{tran} |previous 1/2 words)
$\log p_{LM}$(realization)
$\log p_{LM}$(first word in arg_k^{tran} |previous 1/2 words)
$\log p_{LM}$(realization)/length
$\log p_{LM}$(next word | last 1/2 words in ind_k^{tran})
$\log p_{LM}$(next word | last 1/2 words in arg_k^{tran})

Table 2: Features for abstracts ranking. The language model features are based on a 5-gram language model trained on Gigaword (Graff, 2003) by SRILM (Stolcke, 2002).

5.3 Post-Selection: Redundancy Handling.

Post-selection aims to maximize the information coverage and minimize the redundancy of the summary. Given the generated abstracts A =

Input : relation instances $R = \{\langle ind_i, arg_i \rangle\}_{i=1}^{N}$, generated abstracts $A = \{abs_i\}_{i=1}^{N}$, objective function f , cost function C
Output: final abstract G

$G \leftarrow \Phi$ (empty set);
$U \leftarrow A$;
while $U \neq \Phi$ **do**
  $abs \leftarrow \arg\max_{abs_i \in U} \frac{f(A, G \cup abs_i) - f(A,G)}{C(abs_i)}$;
  if $f(A, G \cup abs) - f(A, G) \geq 0$ **then**
    $G \leftarrow G \cup abs$;
  end
  $U \leftarrow U \setminus abs$;
end

Algorithm 1: Greedy algorithm for post-selection to generate the final summary.

$\{abs_i\}_{i=1}^{N}$, we use a greedy algorithm (Lin and Bilmes, 2010) to select a subset A', where $A' \subseteq A$, to form the final summary. We define w_{ij} as the unigram similarity between abstracts abs_i and abs_j, $C(abs_i)$ as the number of words in abs_i. We employ the following objective function:

$f(A, G) = \sum_{abs_i \in A \setminus G} \sum_{abs_j \in G} w_{i,j}, G \subseteq A$

Algorithm 1 sequentially finds an abstract with the greatest ratio of objective function gain to length, and add it to the summary if the gain is non-negative.

6 Experimental Setup

Corpora. Two disparate corpora are used for evaluation. The AMI meeting corpus (Mccowan et al., 2005) contains 139 scenario-driven meetings, where groups of four people participate in a series of four meetings for a fictitious project of designing remote control. The ICSI meeting corpus (Janin et al., 2003) consists of 75 naturally occurring meetings, each of them has 4 to 10 participants. Compared to the fabricated topics in AMI, the conversations in ICSI tend to be specialized and technical, e.g. discussion about speech and language technology. We use 57 meetings in ICSI and 139 meetings in AMI that include a short (usually one-sentence), manually constructed abstract summarizing each important output for every meeting. Decision and problem summaries are annotated for both corpora. AMI has extra action item summaries, and ICSI has progress summaries. The set of dialogue acts that support each abstract are annotated as such.

System Inputs. We consider two system input settings. In the **True Clusterings** setting, we use the annotations to create perfect partitions of the DAs for input to the system; in the **System Clusterings** setting, we employ a hierarchical agglomerative clustering algorithm used for this task in (Wang and Cardie, 2011). DAs are grouped according to a classifier trained beforehand.

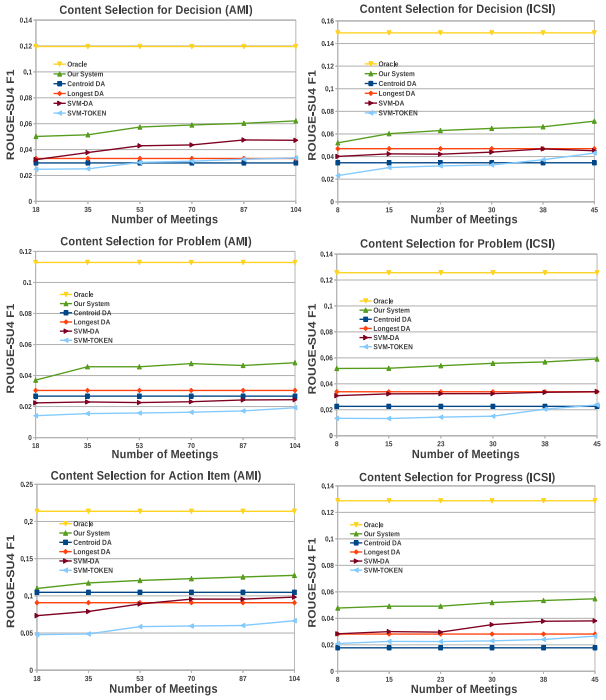

Figure 4: Content selection evaluation by using ROUGE-SU4 (multiplied by 100). SVM-DA and SVM-TOKEN denotes for supervised extract-based methods with SVMs on utterance- and token-level. Summaries for decision, problem, action item, and progress are generated and evaluated for AMI and ICSI (with names in parentheses). X-axis shows the number of meetings used for training.

Baselines and Comparisons. We compare our system with (1) two unsupervised baselines, (2) two supervised extractive approaches, and (3) an oracle derived from the gold standard abstracts.

Baselines. As in Riedhammer et al. (2010), the LONGEST DA in each cluster is selected as the summary. The second baseline picks the cluster prototype (i.e. the DA with the largest TF-IDF similarity with the cluster centroid) as the summary according to Wang and Cardie (2011). Although it is possible that important content is spread over multiple DAs, both baselines allow us to determine summary quality when summaries are restricted to a single utterance.

Supervised Learning. We also compare our approach to two supervised extractive summarization methods — Support Vector Machines (Joachims, 1998) trained with the same fea-

tures as our system (see Table 1) to identify the important **DAs** (no syntax features) (Xie et al., 2008; Sandu et al., 2010) or **tokens** (Fernández et al., 2008) to include into the summary[4].

Oracle. We compute an oracle consisting of the words from the DA cluster that also appear in the associated abstract to reflect the gap between the best possible extracts and the human abstracts.

7 Results

Content Selection Evaluation. We first employ ROUGE (Lin and Hovy, 2003) to evaluate the content selection component with respect to the human written abstracts. ROUGE computes the ngram overlapping between the system summaries with the reference summaries, and has been used for both text and speech summarization (Dang, 2005; Xie et al., 2008). We report ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4) that are shown to correlate with human evaluation reasonably well.

In AMI, four meetings of different functions are carried out in each group[5]. 35 meetings for "conceptual design" are randomly selected for testing. For ICSI, we reserve 12 meetings for testing.

The R-SU4 scores for each system are displayed in Figure 4 and show that our system uniformly outperforms the baselines and supervised systems. The learning curve of our system is relatively flat, which means not many training meetings are required to reach a usable performance level.

Note that the ROUGE scores are relative low when the reference summaries are human abstracts, even for evaluation among abstracts produced by different annotators (Dang, 2005). The intrinsic difference of styles between dialogue and human abstract further lowers the scores. But the trend is still respected among the systems.

Abstract Generation Evaluation. To evaluate the full abstract generation system, the BLEU score (Papineni et al., 2002) (the precision of unigrams and bigrams with a brevity penalty) is computed with human abstracts as reference. BLEU has a fairly good agreement with human judgement and has been used to evaluate a variety of language generation systems (Angeli et al., 2010; Konstas and Lapata, 2012).

[4]We use SVM^{light} (Joachims, 1999) with RBF kernel by default parameters for SVM-based classifiers and regressor.

[5]The four types of meetings in AMI are: project kick-off (35 meetings), functional design (35 meetings), conceptual design (35 meetings), and detailed design (34 meetings).

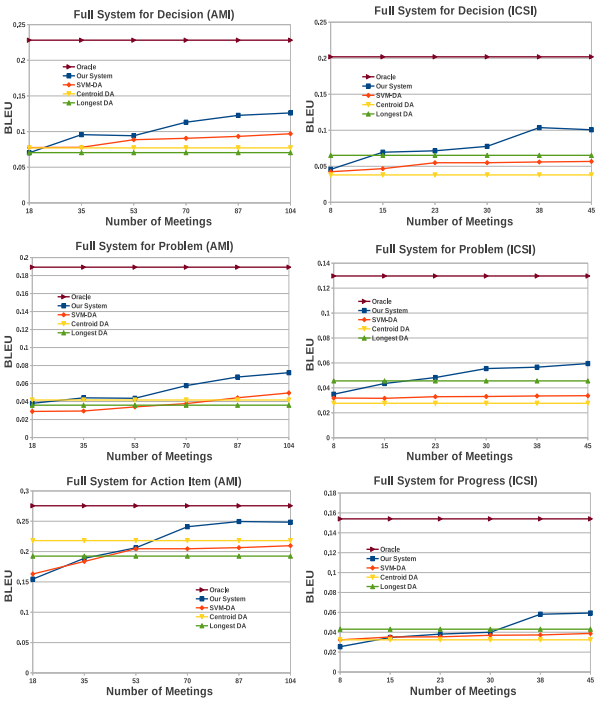

Figure 5: Full abstract generation system evaluation by using BLEU (multiplied by 100). SVM-DA denotes for supervised extractive methods with SVMs on utterance-level.

We are not aware of any existing work generating abstractive summaries for conversations. Therefore, we compare our full system against a supervised utterance-level extractive method based on SVMs along with the baselines. The BLEU scores in Figure 5 show that our system improves the scores consistently over the baselines and the SVM-based approach.

Domain Adaptation Evaluation. We further examine our system in domain adaptation scenarios for decision and problem summarization, where we train the system on AMI for use on ICSI, and vice versa. Table 3 indicates that, with both true clusterings and system clusterings, our system trained on out-of-domain data achieves comparable performance with the same system trained on in-domain data. In most experiments, it also significantly outperforms the baselines and the extract-based approaches ($p < 0.05$).

Human Evaluation. We randomly select 15 decision and 15 problem DA clusters (true clusterings). We evaluate **fluency** (is the text grammatical?) and **semantic correctness** (does the summary convey the gist of the DAs in the cluster?) for OUR SYSTEM trained on IN-domain data

System (True Clusterings)	AMI Decision			ICSI Decision			AMI Problem			ICSI Problem		
	R-2	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*
CENTROID DA	1.3	3.0	7.7	1.8	3.5	3.8	1.0	2.7	4.2	1.0	2.3	2.8
LONGEST DA	1.6	3.3	7.0	2.8	4.7	6.5	1.0	3.0	3.6	1.2	3.4	4.6
SVM-DA (IN)	3.4	4.7	9.7	3.4	4.5	5.7	1.4	2.4	5.0	1.6	3.4	3.4
SVM-DA (OUT)	2.7	4.2	6.6	3.1	4.2	4.6	1.4	2.2	2.5	1.3	3.0	4.6
OUR SYSTEM (IN)	**4.5**	**6.2**	**11.6**	**4.9**	**7.1**	**10.0**	**3.1**	**4.8**	**7.2**	**4.0**	**5.9**	**6.0**
OUR SYSTEM (OUT)	**4.6**	**6.1**	**10.3**	**4.8**	**6.4**	*7.8*	**3.5**	**4.7**	*6.2*	*3.0*	**5.5**	*5.3*
ORACLE	7.5	12.0	22.8	9.9	14.9	20.2	6.6	11.3	18.9	6.4	12.6	13.0
System (System Clusterings)	**AMI Decision**			**ICSI Decision**			**AMI Problem**			**ICSI Problem**		
	R-2	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*	*R-2*	*R-SU4*	*BLEU*
CENTROID DA	1.4	3.3	3.8	1.4	2.1	2.0	0.8	2.8	2.9	0.9	2.3	1.8
LONGEST DA	1.4	3.3	5.7	1.7	3.4	5.5	0.8	3.2	4.1	0.9	3.4	4.4
SVM-DA (IN)	2.6	4.6	10.5	3.5	6.5	7.1	1.8	3.7	4.9	1.8	4.0	4.6
SVM-DA (OUT)	3.4	5.8	10.3	2.7	4.8	6.3	2.1	3.8	4.3	1.5	3.8	3.5
OUR SYSTEM (IN)	*3.5*	*5.4*	*11.7*	**4.4**	**7.4**	**9.1**	**3.3**	**4.6**	**9.5**	*2.3*	*4.2*	**7.4**
OUR SYSTEM (OUT)	**3.9**	**6.4**	*11.4*	**4.1**	*5.1*	**8.4**	**3.6**	**5.6**	**8.9**	*1.8*	*4.0*	**6.8**
ORACLE	6.4	12.0	15.1	8.2	15.2	17.6	6.5	13.0	20.9	5.5	11.9	15.5

Table 3: Domain adaptation evaluation. Systems trained on out-of-domain data are denoted with "(OUT)", otherwise with "(IN)". ROUGE and BLEU scores are multiplied by 100. Our systems that statistically significantly outperform all the other approaches (except ORACLE) are in **bold** ($p < 0.05$, paired t-test). The numbers in *italics* show the significant improvement over the baselines by our systems.

System	Fluency		Semantic		Length
	Mean	*S.D.*	*Mean*	*S.D.*	
OUR SYSTEM (IN)	**3.67**	0.85	3.27	1.03	**23.65**
OUR SYSTEM (OUT)	3.58	0.90	3.25	1.16	**24.17**
SVM-DA (IN)	3.36	0.84	3.44	1.26	38.83

Table 4: Human evaluation results of **Fluency** and **Semantic** correctness for the generated abstracts. The ratings are on 1 (worst) to 5 (best) scale. The average **Length** of the abstracts for each system is also listed.

and OUT-of-domain data, and for the utterance-level extraction system (SVM-DA) trained on in-domain data. Each cluster of DAs along with three randomly ordered summaries are presented to the judges. Five native speaking Ph.D. students (none are authors) performed the task.

We carry out an one-way Analysis of Variance which shows significant differences in score as a function of system ($p < 0.05$, paired t-test). Results in Table 4 demonstrate that our system summaries are significantly more compact and fluent than the extract-based method ($p < 0.05$) while semantic correctness is comparable.

The judges also **rank** the three summaries in terms of the overall quality in content, conciseness and grammaticality. An inter-rater agreement of Fleiss's $\kappa = 0.45$ (moderate agreement (Landis and Koch, 1977)) was computed. Judges selected our system as the best system in 62.3% scenarios (IN-DOMAIN: 35.6%, OUT-OF-DOMAIN: 26.7%). Sample summaries are exhibited in Figure 6.

Decision Summary:
Human: The remote will have push buttons outside, and an LCD and spinning wheel inside.
Our System (In): The group decide to use an LCD display with a spinning wheel. There will be push-buttons on the outside.
Our System (Out): LCD display is going to be with a spinning wheel. It is necessary having push-buttons on the outside.
SVM-DA: Looking at what we've got, we we want an LCD display with a spinning wheel. Just spinning and not scrolling, I would say. I think the spinning wheel is definitely very now. We're having push-buttons on the outside

Problem Summary:
Human: How to incorporate a fruit and vegetable theme into the remote.
Our System (In): Whether to include the shape of fruit. The team had to thinking bright colors.
Our System (Out): It is unclear that the buttons being in the shape of fruit.
SVM-DA: and um Im not sure about the buttons being in the shape of fruit though.

Figure 6: Sample decision and problem summaries generated by various systems for examples in Figure 1.

8 Conclusion

We presented a domain-independent abstract generation framework for focused meeting summarization. Experimental results on two disparate meeting corpora show that our system can uniformly outperform the state-of-the-art supervised extraction-based systems in both automatic and manual evaluation. Our system also exhibits an ability to train on out-of-domain data to generate abstracts for a new target domain.

9 Acknowledgments

This work was supported in part by National Science Foundation Grant IIS-0968450 and a gift from Boeing. We thank Moontae Lee, Myle Ott, Yiye Ruan, Chenhao Tan, and the ACL reviewers for valuable suggestions and advice on various aspects of this work.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 502–512, Stroudsburg, PA, USA. Association for Computational Linguistics.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 16–23, Stroudsburg, PA, USA. Association for Computational Linguistics.

Trung H. Bui, Matthew Frampton, John Dowding, and Stanley Peters. 2009. Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '09, pages 235–243, Stroudsburg, PA, USA. Association for Computational Linguistics.

Giuseppe Carenini, Gabriel Murray, and Raymond Ng. 2011. *Methods for Mining and Summarizing Text Conversations*. Morgan & Claypool Publishers.

Harr Chen, Edward Benson, Tahira Naseem, and Regina Barzilay. 2011. In-domain relation discovery with meta-constraints via posterior regularization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 530–540, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hoa T. Dang. 2005. Overview of DUC 2005. In *Document Understanding Conference*.

Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, July.

Raquel Fernández, Matthew Frampton, John Dowding, Anish Adukuzhiyil, Patrick Ehlen, and Stanley Peters. 2008. Identifying relevant phrases to summarize decisions in spoken meetings. In *INTERSPEECH*, pages 78–81.

Michel Galley. 2006. A skip-chain conditional random field for ranking meeting utterances by importance. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 364–372, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Graff. 2003. English Gigaword.

Michael Heilman and Noah A. Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 609–617, Stroudsburg, PA, USA. Association for Computational Linguistics.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The icsi meeting corpus. volume 1, pages I–364–I–367 vol.1.

Thorsten Joachims. 1998. Text categorization with suport vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK. Springer-Verlag.

Thorsten Joachims. 1999. Advances in kernel methods. chapter Making large-scale support vector machine learning practical, pages 169–184. MIT Press, Cambridge, MA, USA.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ioannis Konstas and Mirella Lapata. 2012. Concept-to-text generation via discriminative reranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 369–378, Stroudsburg, PA, USA. Association for Computational Linguistics.

J R Landis and G G Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Hui Lin and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 912–920, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Fei Liu and Yang Liu. 2009. From extractive to abstractive meeting summaries: can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 261–264, Stroudsburg, PA, USA. Association for Computational Linguistics.

I. Mccowan, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. 2005. The ami meeting corpus. In *In: Proceedings Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research. L.P.J.J. Noldus, F. Grieco, L.W.S. Loijens and P.H. Zimmerman (Eds.), Wageningen: Noldus Information Technology*.

Gabriel Murray, Steve Renals, and Jean Carletta. 2005. Extractive summarization of meeting recordings. In *INTERSPEECH*, pages 593–596.

Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2010a. Interpretation and transformation for abstracting conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 894–902, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gabriel Murray, Giuseppe Carenini, and Raymond T. Ng. 2010b. Generating and validating abstracts of meeting conversations: a user study. In *INLG*.

S. B. Needleman and C. D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, March.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge University Press, New York, NY, USA.

Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tür. 2010. Long story short - global unsupervised models for keyphrase based meeting summarization. *Speech Commun.*, 52(10):801–815, October.

Oana Sandu, Giuseppe Carenini, Gabriel Murray, and Raymond Ng. 2010. Domain adaptation to summarize human conversations. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, pages 16–22, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alex J. Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August.

Andreas Stolcke. 2002. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 2, pages 901–904, Denver, USA.

Marilyn A. Walker, Owen Rambow, and Monica Rogati. 2001. Spot: a trainable sentence planner. In *Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lu Wang and Claire Cardie. 2011. Summarizing decisions in spoken meetings. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, WASDGML '11, pages 16–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lu Wang and Claire Cardie. 2012. Focused meeting summarization via unsupervised relation extraction. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, SIGDIAL '12, pages 304–313, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lusheng Wang and Tao Jiang. 1994. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337–348.

Shasha Xie, Yang Liu, and Hui Lin. 2008. Evaluating the effectiveness of features and sampling in extractive meeting summarization. In *in Proc. of IEEE Spoken Language Technology (SLT*.