Towards Strict Sentence Intersection: Decoding and Evaluation Strategies

Kapil Thadani and **Kathleen McKeown**
Department of Computer Science
Columbia University
New York, NY 10027, USA
{kapil,kathy}@cs.columbia.edu

Abstract

We examine the task of strict sentence intersection: a variant of sentence fusion in which the output must only contain the information present in all input sentences and nothing more. Our proposed approach involves alignment and generalization over the input sentences to produce a generation lattice; we then compare a standard search-based approach for decoding an intersection from this lattice to an integer linear program that preserves aligned content while minimizing the disfluency in interleaving text segments. In addition, we introduce novel evaluation strategies for intersection problems that employ entailment-style judgments for determining the validity of system-generated intersections. Our experiments show that the proposed models produce valid intersections a majority of the time and that the segmented decoder yields advantages over the search-based approach.

1 Introduction

In recent years, there has been growing interest in text-to-text generation problems which transform text according to specifications. Tasks such as sentence compression, which strives to retain the most salient content of an input sentence, and sentence fusion, which attempts to combine the important content in related sentences, are useful components for tackling larger natural language problems such as abstractive summarization of documents. Systems for these types of text-to-text problems are typically evaluated on the informativeness of the output text as judged by human annotators.

A natural aspect of most text generation systems is that a given input can map to a range of lexically diverse outputs. However, text-to-text tasks defined with vague criteria such as the preservation of the "important" information in text can also permit outputs that are *semantically* distinct. This can make evaluation difficult; for instance, system-generated sentences may differ (partially or completely) in informational content from reference human-annotated text. This phenomenon has been noted and discussed in the task of pairwise sentence fusion (Daumé III and Marcu, 2004) and also in sentence compression (McDonald, 2006). Some examples are listed in Table 1.

In this work, we examine the task of *sentence intersection*: a variant of sentence fusion that does not permit semantic variation in the output. A strict[1] intersection system is expected to produce a fused sentence that contains all the information common to its input sentences and *avoid* information that is in just one of the inputs. In other words, a valid intersection should only contain information that is substantiated by all input sentences. The set-theoretic notions of intersection (along with union) have been employed to describe variants of sentence fusion tasks in previous work (Marsi and Krahmer, 2005; Krahmer et al., 2008) but, to our knowledge, this work is the first to explicitly tackle and evaluate the strict intersection task.

We focus on the case of unsupervised pairwise sentence intersection and propose a strategy to yield

[1]We use the term *strict* to make explicit the distinction from traditional fusion systems, which generally aim at notions of intersection but are not formally evaluated with respect to it.

Workshop on Monolingual Text-To-Text Generation, pages 43–53,

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 43–53,
Portland, Oregon, 24 June 2011.

(a) Fusion example from Daumé III and Marcu (2004)	(i) After years of pursuing separate and conflicting paths, AT&T and Digital Equipment Corp. agreed in June to settle their computer-to-PBX differences. (ii) The two will jointly develop an applications interface that can be shared by computers and PBXs of any stripe.
Human fusion #1	*AT&T and Digital Equipment Corp. agreed* in June to settle their computer-to-PBX differences and *develop an applications interface that can be shared by any computer or PBX.*
Human fusion #2	After years of pursuing different paths, *AT&T and Digital agreed to jointly develop an applications interface that can be shared by computers and PBXs of any stripe.*
(b) Compression example from McDonald (2006)	TapeWare , which supports DOS and NetWare 286 , is a value-added process that lets you directly connect the QA150-EXAT to a file server and issue a command from any workstation to back up the server
Human compression #1	*TapeWare* supports DOS and NetWare 286
Human compression #2 (hypothesized)	*TapeWare* lets you connect the QA150-EXAT to a file server

Table 1: Examples of text-to-text generation problems with multiple valid human-generated outputs that differ significantly in semantic content. Italicized text is used to indicate fragments that are semantically identical.

valid intersections that follows the basic framework of previous unsupervised fusion systems (Barzilay and McKeown, 2005; Filippova and Strube, 2008b). In our approach, the input sentences are first aligned using a modified version of a recent phrase-based alignment approach (MacCartney et al., 2008). We assume the alignments that are produced define aspects of the input that must appear in the output fusion and consider decoding strategies to recover intersections that preserve these alignments. In addition to a search-based decoding strategy, we propose a constrained integer linear programming (ILP) formulation that attempts to decode the most fluent sentence covering all these aspects while minimizing the size and disfluency of interleaving text. This is a fairly general model which can also be extended to other alignment-based tasks such as pairwise union and difference.

As this is a substantially more constrained task than generic sentence fusion, we also present a novel evaluation approach that avoids out-of-context salience judgments. We make use of a recently-released corpus of fusion candidates (McKeown et al., 2010) and propose a crowdsourced entailment-style evaluation to determine the *validity* of generated intersections, as well as the grammaticality of the sentences produced. Additionally, automated machine translation (MT) metrics are explored to quantify the amount of information missing from valid intersections. Our decoding strategies show promise under these experiments and we discuss potential directions for improving intersection performance.

2 Related Work

The distinction between intersection and union of text was introduced in the context of sentence fusion (Krahmer et al., 2008; Marsi and Krahmer, 2005) in order to distinguish between traditional fusion strategies that attempted to include only common content and fusions that attempted to include all non-redundant content from the input. We focus here on *strict* sentence intersection, explicitly incorporating a constraint that requires that a produced fusion must not contain information that is not present in all input sentences. This distinguishes our approach from traditional sentence fusion approaches (Jing and McKeown, 2000; Barzilay and McKeown, 2005; Filippova and Strube, 2008b) which generally attempt to retain common information but are typically evaluated in an abstractive summarization context in which additional information in the fusion output does not negatively impact judgments.

This task is also related to the field of sentence compression which has received much attention in recent years (Turner and Charniak, 2005; McDonald, 2006; Clarke and Lapata, 2008; Filippova and Strube, 2008a; Cohn and Lapata, 2009; Marsi et al., 2010). Intersections can be viewed as *guided* compressions in which the redundancy of information content across input sentences in a multidocument setting is assumed to directly indicate its salience, thereby consigning it to the output.

Additionally, in this work, we frequently consider the sentence intersection task from the perspective of textual entailment (cf. §5.1). The textual entailment task involves automatically determining whether a given hypothesis can be inferred from a textual premise (Dagan et al., 2005; Bar-Haim et al., 2006). Automatic construction of positive and negative entailment examples has been explored in the past (Bensley and Hickl, 2008) to provide training data for entailment systems; however the production of text that is simultaneously entailed by two (or more) sentences is a far more constrained and difficult challenge.

ILP has been used extensively for text-to-text generation problems in recent years (Clarke and Lapata, 2008; Filippova and Strube, 2008b; Woodsend et al., 2010), including techniques which incorporate syntax directly into the decoding to imporove the fluency of the resulting text. In this paper, we focus on generating valid intersections and do not incorporate syntactic and semantic constraints into our ILP models; these are areas we intend to explore in the future.

3 The Intersection Task

The need for strict variants of fusion is motivated by considerations of evaluation and utility in text-to-text generation tasks. Without explicit constraints on the semantic content of valid output, the operational definition of fusion can encompass the full spectrum from sentence intersection to sentence union. This makes the comparison of different fusion systems dependent on task-based utility[2]. In addition, intersection comprises an interesting problem in its own right. It necessitates the use of generalization over phrases in order to convey only the content of the input sentences when different wording is used and therefore involves more than just word deletion.

The analogy to set-theoretic intersection in this task implies an underlying consideration of each sentence as a set of informational concepts, similar to previous work in summarization and redundancy (Filatova and Hatzivassiloglou, 2004; Thadani and McKeown, 2008). While we don't commit to any semantic representation for such elements of information, we can nevertheless attempt to *identify* repeated information using well-studied natural language analysis techniques such as alignment and paraphrase recognition, and furthermore *isolate* this information through text-to-text generation techniques.

Consider, for example, the first sentence pair from the examples in Table 2. A valid intersection for these sentences must not contain any information that is not substantiated by both of them, so a fusion that mentions "Mr Litvinenko's poisoning", "Britain" or "Sunday" would not satisfy this criterion. In other words, a valid intersection must necessarily be textually entailed by every input sentence. Following this, we can interpret the sentence intersection task as one that requires the generation of fluent text that is *mutually entailed* by all input sentences[3]. We use this perspective in developing an evaluation technique for strict intersection in §5.1.

A major distinguishing factor between this work and previous work on fusion is that simply adding or deleting words in a sentence is not adequate; in many cases, intersections require additional words or phrases to be introduced in order to generalize over related but non-interchangeable aligned terms (such as "go" and "expand"). Additionally, we must attempt to avoid introducing additional content-bearing text in the output while simultaneously striving to maintain the fluency of text.

3.1 Dataset

A corpus of sentence fusion instances was recently made available by McKeown et al. (2010), consisting of 297 sentence pairs taken from newswire clusters and manually judged as being good candidates for fusion. Each sentence pair is accompanied by human-produced intersections and unions collected via Amazon's Mechanical Turk service[4]. McKeown et al. (2010) noted that union responses are mostly valid but intersections are frequently incorrect and

[2]For instance, systems may trade off conciseness against grammaticality, or informational content with degree of support across the input sentences.

[3]From this perspective, the complementary task of sentence union involves the generation of fluent text that entails all the input sentences.

[4]`http://www.mturk.com`

1	(i) Home Secretary John Reid said Sunday the inquiry would go wherever "the police take it." (ii) It comes as Home Secretary John Reid said the inquiry into Mr Litvinenko's poisoning would expand beyond Britain.
2	(i) Traces of polonium have been found on the planes on which they are believed to have travelled between London and Moscow. (ii) Small traces of radioactive substances had been found on the planes.
3	(i) Prosecutors allege that the accuser, who appeared in the program, was molested after the show aired. (ii) Prosecutors allege that the boy, a cancer survivor, was molested twice after the program aired.

Table 2: Example sentence pairs from the McKeown et al. (2010) corpus. Table 3 contains the corresponding system-generated intersections for these sentence pairs.

hypothesized that the task is more confusing for untrained annotators. A similar phenomenon was noted by Krahmer et al. (2008): while demonstrating that query-based human fusions exhibited less variation than generic fusions, it was also observed that intersections varied more than unions.

Due to the absence of adequate training data for intersection, our approach to the task is unsupervised, similar to previous work in fusion (Barzilay and McKeown, 2005; Filippova and Strube, 2008b) and sentence compression (Clarke and Lapata, 2008; Filippova and Strube, 2008a). Additionally, we focus on the case of pairwise sentence intersection and assume that the common information between the input sentence pair can be represented within a single output sentence. As a result, although the McKeown et al. (2010) corpus cannot be used for training an intersection model, we can make use of the sentence pairs it contains for evaluation.

4 Models for intersection

Our proposed strategies for sentence intersection involve phrase-based alignment, intermediate generalization steps that build a generation lattice and techniques for decoding an output sentence, as described below.

4.1 Phrase-based alignment

The alignment phase is a major component of any intersection system as it is used to uncover the common segments in the input that must be preserved in the output. We make use of an adaptation of the supervised MANLI phrase-based alignment technique originally developed for textual entailment systems (MacCartney et al., 2008); our implementation replaces approximate search-based decoding with exact ILP-based alignment decoding and incorporates syntactic constraints to produce more precise alignments (Thadani and McKeown, 2011). The aligner is trained on a corpus of human-generated alignment annotations produced by Microsoft Research (Brockett, 2007) for inference problems from the second Recognizing Textual Entailment (RTE2) challenge (Bar-Haim et al., 2006).

Entailment problems are inherently asymmetric because premise text is generally larger than hypothesis text; however, this does not apply to our intersection problems and consequently our MANLI implementation drops asymmetric indicator features. The absence of these features impacts alignment performance on RTE2 data but our reimplementation performs comparably to the original model under the alignment evaluation from MacCartney et al. (2008).

4.2 Ontology-based generalization

An aligned phrase pair produced by the previous step does not necessarily indicate that the phrases are equivalent but merely that they are similar in the given sentence context (such as "accuser" and "boy" in the third example from Table 2). We need to generalize over these phrases as they are not interchangeable from the perspective of the intersection task. We consider an alignment as containing three types of aligned phrases:

1. **Identical phrases or paraphrases:** Either of these may appear in the output
2. **Entailed phrases:** Only the entailed phrase must appear in a valid intersection
3. **Instances of a general concept:** The common concept must be lexicalized in the output

Although generalization of words within standalone sentences is usually hampered by word sense ambiguity, our approach is less likely to encounter this problem because we can generalize *simultaneously* over phrases which have already been aligned using additional information (such as their neighboring context), thus avoiding generalizations that do not fit the alignment.

For our experiments, we make use of the Wordnet ontology (Miller, 1995) to find the hypernyms common to every aligned pair of non-identical phrases, and only attempt to detect entailments which are comprised of specific instances that entail general concepts. This approach can be augmented by the use of entailment corpora and distributional clustering which we intend to explore in future work. We also use the lexical resource CatVar (Habash and Dorr, 2003) to try to generate morphological variants of aligned words that enable them to be interchanged without creating disfluencies.

4.3 Pragmatic abstraction

Our strategy assumes that aligned text must be preserved in output intersections whereas unaligned text must be minimized. However, unaligned text cannot simply be dropped as it may contain vital portions for generating fluent text. In addition, unaligned phrases can be caused by paraphrased or metaphorical text that the aligner is not capable of identifying. For example, the phrases "polonium" and "radioactive substances" in the second sentence pair from Table 2 fail to align with each other.

On the other hand, retaining unaligned text from one of the input sentences for the sake of fluency is likely to introduce information that is not supported by the other input sentence. We therefore need to abstract away as much content from the unaligned portions of the text as possible. For this purpose, we generate a large number of potential compressions and abstractions for every unaligned span that occurs between two consecutive aligned phrases in each sentence. These compressions and abstractions, referred to as *interleaving paths*, between pairs of aligned phrases essentially construct a lattice over the input sentences that encodes all potential intersection outputs.

Generation of interleaving paths is accomplished through the application of rules on the dependency parse structure over unaligned text spans from a single sentence (as well as spans that occur before the first aligned phrase and after the last aligned phrase in each sentence). Interleaving paths are generated by applying rules that:

1. Drop insignificant dependent words and unaligned prepositional phrases
2. Replace content-bearing verbs with tense-adjusted generic variants such as "did something" and "happened", with an exception for statement verbs
3. Replace nouns with generic words such as "someone" or "something", using Wordnet to determine which generic variant fits a noun
4. Suggest connective text fragments such as "something about" to cover long spans and clause boundaries

Our abstraction rules are relatively simple but can often generate reasonable interleaving paths. In general, we note that shorter abstractions are less likely to include glaring grammatical errors because long unaligned spans are often indicative of problematic alignments that either incorrectly relate unconnected terms or fail to recognize paraphrases.

4.4 Decoding strategies

After sentence alignment, generalization over aligned phrases and the construction of interleaving paths, we are left with a lattice that encodes potential intersections of the input sentence. Figure 1 describes the general structure of this lattice. Every alignment link encompasses a set of aligned phrases. Phrases may be identical or generalizations, in which case they can appear in the context of either sentence, or they may be sentence-specific (for example, verbs with different tenses or nominalizations like "nominated" and "nominations"). Additionally, the abstraction phase generates interleaving paths from unaligned spans between all pairs of alignment links. These paths are generated from individual sentences and can only be used to connect phrases that appear in the context of those sentences.

Our task now reduces to recovering a well-formed intersection from this lattice. We make use of a language model (LM) to judge fluency and propose two techniques to decode high-scoring text from the lattice: a simple beam-search technique and an ILP

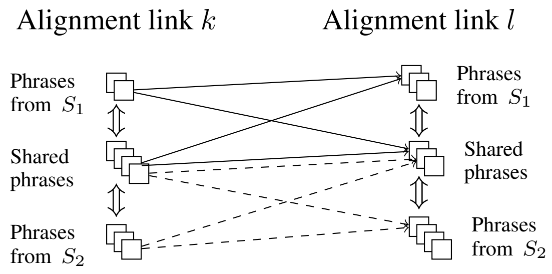

Figure 1: The general structure of one segment of the alignment lattice, illustrating the potential interleaving paths between aligned phrases. Solid lines indicate paths derived from sentence 1 and dashed lines indicate paths derived from sentence 2

strategy that leverages our initial assumption that all aligned phrases must appear in the output.

4.4.1 Beam search

Search-based decoding is often employed in phrase-based MT systems (Och and Ney, 2003) and is implemented in the Moses toolkit[5]; similar approaches have also been used in text-to-text generation tasks (Barzilay and McKeown, 2005; Soricut and Marcu, 2006). This technique attempts to find the highest-scoring sentence string under the LM by unwrapping and searching through a lattice. Since the dynamic programming search could require an exponential number of search states, a fixed-width beam can be used to control the number of search states being actively considered at each step.

In order to decode an intersection problem, we first pick a beam size B and initialize the list of candidate search states with the first interleaving paths in each sentence. At every iteration, we consider the B candidates with the highest normalized scores under the LM and remove them from the candidate list. Each candidate is then *advanced*, i.e., all aligned phrases and interleaving paths following it are examined, scored and added to the candidate list. We continue searching in this manner until B candidates have covered all aligned phrases; the highest scoring candidate is then retrieved as the target intersection.

4.4.2 Segmented decoding

While beam search is a viable strategy for decoding intersections, its performance is contingent on the beam size parameter and it is not guaranteed to return the highest scoring sentence under the LM. For instance, if a potential intersection starts with unusual text, it is unlikely to be explored by the search-based approach even if it is the optimal solution to the decoding problem. To address this, we also propose an alternative decoding problem that can be formulated as the optimization of a linear objective function with linear constraints. This can then be solved exactly by well-studied algorithms using off-the-shelf ILP solvers[6].

This decoding problem does not look for the highest scoring sentence under the LM; instead, it attempts to find the set of interleaving paths and aligned phrases that are most locally coherent[7] under the LM. Good phrase-path combinations that occur towards the tail end of an intersection can thus be put on even footing with the combinations that appear in the beginning. Although the two problems consider different objective functions, they are both engaged in the same overall goal: that of recovering a fluent sentence from the lattice.

We first define boolean indicator variables $a_i^k \in A_k$ for every aligned phrase in each aligned link A_k present in the intersection problem $\mathcal{I}$. We also introduce indicator variables p_{ij}^{kl} for every possible interleaving path between aligned phrases a_i^k and a_j^l. The linear objective for $\mathcal{I}$ that maximizes the local coherence of all phrases can be expressed as

$$f = \max \sum_{A_k, A_l \in \mathcal{I}} \sum_{i=0}^{|A_k|} \sum_{j=0}^{|A_l|} p_{ij}^{kl} \times score(p_{ij}^{kl})$$

where $score(p_{ij}^{kl})$ is the normalized LM score of the fragment of text representing $a_i^k \; p_{ij}^{kl} \; a_j^l$. In other words, the score for each interleaving path is calculated by appending it and the two phrases it connects into a single fragment of text and determining the score of that fragment under an LM[8].

[5] http://www.statmt.org/moses/

[6] We use LPsolve: http://lpsolve.sourceforge.net/

[7] As noted by Clarke and Lapata (2008), normalizing LM scores cannot be easily accomplished with linear constraints and we do not have training data to devise appropriate word-insertion penalties as used in MT.

[8] If the fragment of text is smaller than the LM size, we consider additional sentence context around the aligned phrases rather than backing off to a smaller LM size to avoid a bias towards short but ungrammatical interleaving paths.

We now introduce linear constraints to keep the problem well-formed. First, we add a restriction to ensure that only one phrase from each alignment link is present in the solution.

$$\sum_{a_i^k \in A_k} a_i^k = 1 \qquad \forall A_k \in \mathcal{I}$$

We can also ensure that interleaving paths are only in the solution when the aligned phrases that they connect together are themselves present using the following set of constraints.

$$\begin{aligned}
&a_i^k - \sum_{i=0}^{|A_k|} p_{i*}^{k*} = 1 \qquad \forall a_i^k \in A_k, A_k \in \mathcal{I} \\
&a_j^l - \sum_{j=0}^{|A_l|} p_{*j}^{*l} = 1 \qquad \forall a_j^l \in A_l, A_l \in \mathcal{I} \\
&p_{ij}^{kl} - a_i^k <= 0 \qquad \forall i, j, k, l \\
&p_{ij}^{kl} - a_j^l <= 0 \qquad \forall i, j, k, l
\end{aligned}$$

As we don't restrict the structure of the lattice in any way and allow crossing alignment links, the program as defined thus far is capable of generating cyclic and fragmented solutions. To combat this, we add dummy start and end phrase variables and introduce additional *single commodity flow* constraints (Magnanti and Wolsey, 1994) adapted from Martins et al. (2009) over the interleaving paths to guarantee that the output will only involve a linear sequence of aligned phrases and paths.

5 Evaluation

We now turn to the design of experiments for the strict sentence intersection task and discuss the performance of the proposed models using the corpus provided by McKeown et al. (2010). We use a beam size of 50 for the beam search decoder and a 4-gram LM for all experiments. Dependency parsing is accomplished with MICA, a TAG-based parser (Bangalore et al., 2009). Our primary considerations for studying system-generated fusions are *validity* (whether the output contains only the information common to each sentence), *coverage* (whether the output contains all the common information in the input sentences) and the *fluency* of the output.

5.1 Evaluating Validity and Fluency

Evaluating the validity of an intersection involves determining whether it contains only the information contained in each sentence and nothing else. In order to do this, we make use of the interpretation of valid intersections as being mutually entailed by the input sentences. It follows that the task of judging the validity of an intersection can simply be decomposed into two tasks that judge whether the intersection is entailed by each input sentence.

We make use of Amazon's Mechanical Turk (AMT) platform to have humans evaluate the intersections produced. Crowdsourcing annotations and judgments in this manner has been shown to be cheap and effective for natural language tasks (Snow et al., 2008) and has recently been employed in similar entailment-detection tasks (Negri and Mehdad, 2010; Buzek et al., 2010). Since we only seek judgments on produced intersections and avoid presenting both input sentences to users, we do not anticipate the noisiness that was noted by McKeown et al. (2010) when asking AMT users to *generate* intersections.

Each entailment task is framed as a multiple choice question. An AMT user is shown just one input sentence (the *premise* in entailment terminology) along with a potential intersection (the *hypothesis*) and is required to respond to whether there is any new or different information in the latter that is not in the former. They can respond on a 3-point scale (*yes/no/maybe*) where *maybe* is clarified to include ambiguous rewording in the intersection. For a given intersection instance, the responses[9] using each input sentence as the premise are averaged separately and then combined[10] to give a measure of how well the intersection is entailed by *both* sentences.

A second question allows the user to specify the grammaticality of the intersection on a 4-point scale. As this measure doesn't depend on the input sentence presented to the AMT user, all scores provided are simply averaged per intersection.

[9]Each instance is presented to 6 AMT users, 3 per premise. Responses were automatically filtered for spam and removing the largest outlier from each per-premise or per-intersection group did not yield a notable change in relative performance.

[10]We use the harmonic mean for combination, but the results are largely similar when using an arithmetic mean.

	Intersection output	Fluency	Validity
Aligned words	(i) Home Secretary John Reid said the inquiry would go.	0.667	0.800
	(ii) Home Secretary John Reid said the inquiry would expand.	0.778	
Beam search	Home Secretary John Reid said something about the inquiry would move wherever "the something take it".	0.389	0.667
Segmented decoder	Home Secretary John Reid said the inquiry would change.	0.944	0.909
Aligned words	(i) Traces of have been found on the planes.	0.445	1.000
	(ii) traces of had been found on the planes.	0.556	
Beam search	Small traces of some things have been found on the planes.	0.611	0.909
Segmented decoder	Small traces of had been found on the planes.	0.500	0.741
Aligned words	(i) Prosecutors allege that the accuser the program was molested after aired.	0.167	0.800
	(ii) Prosecutors allege that the boy was molested after the program aired.	1.000	
Beam search	Prosecutors allege that the being, who did something in the program, was molested after something about aired.	0.400	0.909
Segmented decoder	Prosecutors allege that the organism, who did something, was molested after the program aired.	0.667	0.857

Table 3: Intersections produced for the examples introduced in Table 2 along with judgments from AMT users.

	Validity	Fluency	Har. Mean
Other sentence	0.188	0.945	0.314
Aligned words	0.863	0.563†	0.682†
Beam search	0.729	0.450	0.557
Segmented decoder	0.812†	0.504	0.622
Oracle combination	0.813†	0.575†	0.674†

Table 4: Results of the AMT evaluation described in §5.1. Statistically insignificant differences within columns are indicated with †; all other entries are significantly distinct at $p \leq 0.05$.

5.2 Results of AMT evaluation

Table 4 contains the results from this evaluation over the McKeown et al. (2010) corpus[11] and Table 3 shows the system-produced intersections corresponding to the examples from §3. We report normalized scores of validity and fluency for ease of comparison, as well as their unweighted harmonic mean as a crude measure of combined human judgment. In addition to the beam search and segmented decoders, we report the performance of two upper-bound systems that present artificial hypothesis sentences to AMT users. *Other sentence* is simply the sentence that is not the current premise from the sentence pair; although this is rarely an appropriate intersection in the data, it is useful as a measure of how well humans judge grammaticality and information content. *Aligned words* is the aligned subset of the premise sentence; this is quite likely to be considered a valid entailment by AMT users as no new words are introduced. Although the latter also scores surprisingly well on fluency, we must note that this is not an actual intersection solution: the aligned words displayed to AMT users for a given intersection instance are different depending on which input sentence is displayed as the premise.

Turning to the systems under study, we observe that the ILP-based segmented decoder produces text that is judged more fluent on average than the beam search decoder. In order to judge the degree of overlap between the two systems, we also report the performance of a pseudo-hybrid *oracle combination* system which assumes the presence of an oracle that runs both decoders and always chooses the output intersection that is more grammatical. The improved performance illustrates that each decoder has its advantages and that a real hybrid system might yield improvements over either approach.

5.3 Evaluating Coverage

While validity experiments test whether the proposed intersections contain extraneous or unsupported information, we also need to check whether the intersections contain *all* the information that is shared between the input sentences. This cannot be factored into a task that involves only one input sentence and therefore cannot be easily accomplished

[11] The first 20 sentence pairs of the corpus were examined when devising abstraction rules and are therefore excluded from these results.

	BLEU	NIST
Aligned words	0.682	11.10
Beam search	0.726	10.53
Segmented decoder	0.818	11.56

Table 5: Results of the automated evaluation for coverage of intersections described in §5.3.

without annotators who understand the concept of intersection.

We instead attempt to utilize the high-quality human-generated union dataset from McKeown et al. (2010) in evaluating the coverage of our intersection systems. Using the simple absorption law $A \cap (A \cup B) = A$, we assume that the coverage of intersection systems can be judged by how well they can recover an input sentence from human-generated unions. The resulting outputs are compared to the original input sentences in an MT-style evaluation under two commonly-used metrics: BLEU (Papineni et al., 2002) and NIST (Doddington, 2002).

The results of this automated evaluation are shown in Table 5. The *aligned words* system here always considers words from the union sentence and can therefore be seen as a baseline system. We observe that the segmented decoder produces output that is judged most similar to the input sentences under BLEU, which measures n-gram overlap, although results under NIST (which gives additional weight to *rarer* n-grams) are less conclusive.

6 Discussion

The experimental results indicate that the two systems we describe, particularly the segmented decoder, do a reasonable job of finding valid intersections with good coverage; however, producing fluent output remains a challenge. Analysis of the intersections produced leads us to note that the quality of interleaving paths is the prime obstacle to improving intersection output (cf. Table 3): producing syntactically-valid textual abstractions to connect text is a challenge that is not met by our simple rule-based approach. Furthermore, we notice that the quality of alignment also factors in to this problem: systems that miss phrases which should be aligned or systems that mistakenly align faraway fragments both cause spans of unaligned text that must be then abstracted over.

We hypothesize that these issues could be tackled with the use of joint models: a system that aligns as it decodes could reduce the need for abstraction over long unaligned spans, although care would have to be taken to ensure that coverage is maintained. Additionally, richer lexical resources such as wider-coverage ontologies (Snow et al., 2006) and entailment/paraphrase dictionaries could aid in improving coverage. Finally, previous work in fusion (Filippova and Strube, 2008b; Filippova and Strube, 2009) has noted that models based on syntax outperform techniques that rely solely on LM scores to determine fluency, and strict intersection appears to be well-suited for further exploration in this vein.

7 Conclusion

We have examined the text-to-text generation task of strict sentence intersection, which restricts semantic variation in the output and necessarily invokes the problems of generalization and abstraction in addition to the usual challenge of producing fluent text. We tackle the task as lattice decoding and discuss two decoding strategies for producing valid intersections. In addition, we assume that strict intersection tasks are best considered as problems of mutual entailment generation and describe evaluation strategies for this task that make use of both human judgments as well as automated metrics run over a related corpus. Experimental results indicate that these systems are fairly effective at generating valid intersections and that our novel segmented decoder strategy outperforms the traditional beam search approach. Although fluency remains a challenge, we hypothesize that the use of joint models, syntactic constraints and lexical resources could bring improvements.

Acknowledgments

The authors are grateful to the anonymous reviewers for their helpful feedback. This material is based on research supported in part by the U.S. National Science Foundation (NSF) under IIS-05-34871. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

Srinivas Bangalore, Pierre Boullier, Alexis Nasr, Owen Rambow, and Benoît Sagot. 2009. MICA: a probabilistic dependency parser based on tree insertion grammars. In *Proceedings of HLT-NAACL: Short Papers*, pages 185–188.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second PASCAL Recognising Textual Entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Jeremy Bensley and Andrew Hickl. 2008. Unsupervised resource creation for textual inference applications. In *Proceedings of LREC*.

Chris Brockett. 2007. Aligning the 2006 RTE corpus. Technical Report MSR-TR-2007-77, Microsoft Research.

Olivia Buzek, Philip Resnik, and Benjamin B. Bederson. 2010. Error driven paraphrase annotation using mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 217–221.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression: an integer linear programming approach. *Journal of Artifical Intelligence Research*, 31:399–429, March.

Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*.

Hal Daumé III and Daniel Marcu. 2004. Generic sentence fusion is an ill-defined summarization task. In *Proceedings of the ACL Text Summarization Branches Out Workshop*, pages 96–103.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of HLT*, pages 138–145.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. A formal model for information selection in multi-sentence text extraction. In *Proceedings of COLING*, page 397.

Katja Filippova and Michael Strube. 2008a. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, INLG '08, pages 25–32.

Katja Filippova and Michael Strube. 2008b. Sentence fusion via dependency graph compression. In *Proceedings of EMNLP*, pages 177–185.

Katja Filippova and Michael Strube. 2009. Tree linearization in English: improving language model based approaches. In *Proceedings of NAACL*, pages 225–228.

Nizar Habash and Bonnie Dorr. 2003. A categorial variation database for English. In *Proceedings of NAACL*, NAACL '03, pages 17–23.

Hongyan Jing and Kathleen R. McKeown. 2000. Cut and paste based text summarization. In *Proceedings of NAACL*, pages 178–185.

Emiel Krahmer, Erwin Marsi, and Paul van Pelt. 2008. Query-based sentence fusion is better defined and leads to more preferred results than generic sentence fusion. In *Proceedings of ACL*, pages 193–196.

Bill MacCartney, Michel Galley, and Christopher D. Manning. 2008. A phrase-based alignment model for natural language inference. In *Proceedings of EMNLP*, pages 802–811.

Thomas L. Magnanti and Laurence A. Wolsey. 1994. Optimal trees. In *Technical Report 290-94, Massechusetts Institute of Technology, Operations Research Center*.

Erwin Marsi and Emiel Krahmer. 2005. Explorations in sentence fusion. In *Proceedings of the European Workshop on Natural Language Generation*, pages 109–117.

Erwin Marsi, Emiel Krahmer, Iris Hendrickx, and Walter Daelemans. 2010. On the limits of sentence compression by deletion. In Emiel Krahmer and Mariët Theune, editors, *Empirical Methods in Natural Language Generation*, pages 45–66. Springer-Verlag, Berlin, Heidelberg.

André F. T. Martins, Noah A. Smith, and Eric P. Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of ACL-IJCNLP*, pages 342–350.

Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of EACL*, pages 297–304.

Kathleen McKeown, Sara Rosenthal, Kapil Thadani, and Coleman Moore. 2010. Time-efficient creation of an accurate sentence fusion corpus. In *Proceedings of NAACL-HLT*.

George A. Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38:39–41, November.

Matteo Negri and Yashar Mehdad. 2010. Creating a bi-lingual entailment corpus through translations with mechanical turk: $100 for a 10-day rush. In *Proceedings of the NAACL HLT 2010 Workshop on Creating*

Speech and Language Data with Amazon's Mechanical Turk, pages 212–216.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51, March.

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

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of ACL*, pages 801–808.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP*, pages 254–263.

Radu Soricut and Daniel Marcu. 2006. Stochastic language generation using widl-expressions and its application in machine translation and summarization. In *Proceedings of ACL*, pages 1105–1112.

Kapil Thadani and Kathleen McKeown. 2008. A framework for identifying textual redundancy. In *Proceedings of COLING*, pages 873–880.

Kapil Thadani and Kathleen McKeown. 2011. Optimal and syntactically-informed decoding for monolingual phrase-based alignment. In *Proceedings of ACL*.

Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of ACL*, pages 290–297.

Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Title generation with quasi-synchronous grammar. In *Proceedings of EMNLP*, EMNLP '10, pages 513–523.