Sentence Compression with Joint Structural Inference

Kapil Thadani and **Kathleen McKeown**
Department of Computer Science
Columbia University
New York, NY 10025, USA
{kapil,kathy}@cs.columbia.edu

Abstract

Sentence compression techniques often assemble output sentences using fragments of lexical sequences such as n-grams or units of syntactic structure such as edges from a dependency tree representation. We present a novel approach for discriminative sentence compression that unifies these notions and jointly produces sequential *and* syntactic representations for output text, leveraging a compact integer linear programming formulation to maintain structural integrity. Our supervised models permit rich features over heterogeneous linguistic structures and generalize over previous state-of-the-art approaches. Experiments on corpora featuring human-generated compressions demonstrate a 13-15% relative gain in 4-gram accuracy over a well-studied language model-based compression system.

1 Introduction

Recent years have seen increasing interest in text-to-text generation tasks such as paraphrasing and text simplification, due in large part to their direct utility in high-level natural language tasks such as abstractive summarization. The task of sentence compression in particular has benefited from the availability of a number of useful resources such as the the Ziff-Davis compression corpus (Knight and Marcu, 2000) and the Edinburgh compression corpus (Clarke and Lapata, 2006b) which make compression problems highly relevant for data-driven approaches involving language generation.

The sentence compression task addresses the problem of minimizing the lexical footprint of a sentence, i.e., the number of words or characters in it, while preserving its most salient information. This is illustrated in the following example from the compression corpus of Clarke and Lapata (2006b):

> **Original:** In 1967 Chapman, who had cultivated a conventional image with his ubiquitous tweed jacket and pipe, by his own later admission stunned a party attended by his friends and future Python colleagues by coming out as a homosexual.
>
> **Compressed:** In 1967 Chapman, who had cultivated a conventional image, stunned a party by coming out as a homosexual.

Compression can therefore be viewed as analogous to text summarization[1] defined at the sentence level. Unsurprisingly, independent selection of tokens for an output sentence does not lead to fluent or meaningful compressions; thus, compression systems often assemble output text from units that are larger than single tokens such as n-grams (McDonald, 2006; Clarke and Lapata, 2008) or edges in a dependency structure (Filippova and Strube, 2008; Galanis and Androutsopoulos, 2010). These systems implicitly rely on a structural representation of text—as a sequence of tokens or as a dependency tree respectively—to to underpin the generation of an output sentence.

In this work, we present *structured transduction*: a novel supervised framework for sentence compression which employs a joint inference strategy to simultaneously recover sentence compressions under *both* these structural representations of text—a token sequence as well as a tree of syntactic dependencies. Sentence generation is treated as a discriminative structured prediction task in which rich linguistically-motivated

[1] To further the analogy, compression is most often formulated as a word deletion task which parallels the popular view of summarization as a sentence extraction task.

Proceedings of the Seventeenth Conference on Computational Natural Language Learning, pages 65–74,
Sofia, Bulgaria, August 8-9 2013. ©2013 Association for Computational Linguistics

features can be used to predict the informativeness of specific tokens within the input text as well as the fluency of n-grams and dependency relationships in the output text. We present a novel constrained integer linear program that optimally solves the joint inference problem, using the notion of *commodity flow* (Magnanti and Wolsey, 1994) to ensure the production of valid acyclic sequences and trees for an output sentence.

The primary contributions of this work are:

- A supervised sequence-based compression model which outperforms Clarke & Lapata's (2008) state-of-the-art sequence-based compression system without relying on any hard syntactic constraints.
- A formulation to jointly infer tree structures alongside sequentially-ordered n-grams, thereby permitting features that factor over both phrases and dependency relations.

The structured transduction models offer additional flexibility when compared to existing models that compress via n-gram or dependency factorizations. For instance, the use of commodity flow constraints to ensure well-formed structure permits arbitrary reorderings of words in the input and is not restricted to producing text in the same order as the input like much previous work (McDonald, 2006; Clarke and Lapata, 2008; Filippova and Strube, 2008) *inter alia*.[2]

We ran compression experiments with the proposed approaches on well-studied corpora from the domains of written news (Clarke and Lapata, 2006b) and broadcast news (Clarke and Lapata, 2008). Our supervised approaches show significant gains over the language model-based compression system of Clarke and Lapata (2008) under a variety of performance measures, yielding 13-15% relative F_1 improvements for 4-gram retrieval over Clarke and Lapata (2008) under identical compression rate conditions.

2 Joint Structure Transduction

The structured transduction framework is driven by the fundamental assumption that generating fluent text involves considerations of diverse structural relationships between tokens in both input and output sentences. Models for sentence compression often compose text from units that are larger than individual tokens, such as n-grams which describe a token sequence or syntactic relations which comprise a dependency tree. However, our approach is specifically motivated by the perspective that *both* these representations of a sentence—a sequence of tokens and a tree of dependency relations—are equally meaningful when considering its underlying fluency and integrity. In other words, models for compressing a token sequence must also account for the compression of its dependency representation and vice versa.

In this section, we discuss the problem of recovering an optimal compression from a sentence as a linear optimization problem over heterogenous substructures (cf. §2.1) that can be assembled into valid and consistent representations of a sentence (cf. §2.2). We then consider rich linguistically-motivated features over these substructures (cf. §2.3) for which corresponding weights can be learned via supervised structured prediction (cf. §2.4).

2.1 Linear Objective

Consider a single compression instance involving a source sentence S containing m tokens. The notation $\widehat{S}$ is used to denote a well-formed compression of S. In this paper, we follow the standard assumption from compression research in assuming that candidate compressions $\widehat{S}$ are assembled from the tokens in S, thereby treating compression as a word-deletion task. The inference step aims to retrieve the output sentence $\widehat{S}^*$ that is the most likely compression of the given input S, i.e., the $\widehat{S}$ that maximizes $p(\widehat{S}|S) \propto p(\widehat{S}, S)$ or, in an equivalent discriminative setting, the $\widehat{S}$ that maximizes a feature-based score for compression

$$\widehat{S}^* \triangleq \arg\max_{\widehat{S}} \ \mathbf{w}^\top \mathbf{\Phi}(S, \widehat{S}) \tag{1}$$

where $\mathbf{\Phi}(S, \widehat{S})$ denotes some feature map parameterized by a weight vector $\mathbf{w}$.

Let $T \triangleq \{t_i : 1 \leq i \leq m\}$ represent the set of tokens in S and let $x_i \in \{0, 1\}$ represent a token indicator variable whose value corresponds to whether token t_i is present in the output sentence $\widehat{S}$. The incidence vector $\mathbf{x} \triangleq \langle x_1, \ldots, x_m \rangle^\top$ therefore represents an entire token configuration that is equivalent to some subset of T.

If we were to consider a simplistic bag-of-tokens scenario in which the features factor entirely over the tokens from T, the highest-scoring

[2] We do not evaluate token reordering in the current work as the corpus used for experiments in §3 features human-generated compressions that preserve token ordering.

compression under (1) would simply be the token configuration that maximizes a linear combination of per-token scores, i.e., $\sum_{t_i \in T} x_i \cdot \theta_{\text{tok}}(i)$ where $\theta_{\text{tok}} : \mathbb{N} \rightarrow \mathbb{R}$ denotes a linear scoring function which measures the relative value of retaining t_i in a compression of S based on its features, i.e., $\theta_{\text{tok}}(i) \triangleq \boldsymbol{w}_{\text{tok}}^{\top} \boldsymbol{\phi}_{\text{tok}}(t_i)$. Although this can be solved efficiently under compression-rate constraints, the strong independence assumption used is clearly unrealistic: a model that cannot consider any relationship between tokens in the output does not provide a token ordering or ensure that the resulting sentence is grammatical.

The natural solution is to include higher-order factorizations of linguistic structures such as n-grams in the objective function. For clarity of exposition, we assume the use of trigrams without loss of generality. Let U represent the set of all possible trigrams that can be constructed from the tokens of S; in other words $U \triangleq \{\langle t_i, t_j, t_k \rangle : t_i \in T \cup \{\text{START}\}, t_j \in T, t_k \in T \cup \{\text{END}\}, i \neq j \neq k\}$. Following the notation for token indicators, let $y_{ijk} \in \{0, 1\}$ represent a trigram indicator variable for whether the contiguous sequence of tokens $\langle t_i, t_j, t_k \rangle$ is in the output sentence. The incidence vector $\mathbf{y} \triangleq \langle y_{ijk} \rangle_{\langle t_i, t_j, t_k \rangle \in U}$ hence represents some subset of the trigrams in U. Similarly, let V represent the set of all possible dependency edges that can be established among the tokens of S and the pseudo-token ROOT, i.e., $V \triangleq \{\langle i, j \rangle : i \in T \cup \{\text{ROOT}\}, j \in T, t_j \text{ is a dependent of } t_i \text{ in } S\}$. As before, $z_{ij} \in \{0, 1\}$ represents a dependency arc indicator variable indicating whether t_j is a direct dependent of t_i in the dependency structure of the output sentence, and $\mathbf{z} \triangleq \langle z_{ij} \rangle_{\langle t_i, t_j \rangle \in V}$ represents a subset of the arcs from V.

Using this notation, any output sentence $\widehat{S}$ can now be expressed as a combination of some token, trigram and dependency arc configurations $\langle \mathbf{x}, \mathbf{y}, \mathbf{z} \rangle$. Defining θ_{ngr} and θ_{dep} analogously to θ_{tok} for trigrams and dependency arcs respectively, we rewrite (1) as

$$\begin{aligned}
\widehat{S}^* = \operatorname*{arg\,max}_{\mathbf{x},\mathbf{y},\mathbf{z}} & \sum_{t_i \in T} x_i \cdot \theta_{\text{tok}}(i) \\
& + \sum_{\langle t_i, t_j, t_k \rangle \in U} y_{ijk} \cdot \theta_{\text{ngr}}(i, j, k) \\
& + \sum_{\langle t_i, t_j \rangle \in V} z_{ij} \cdot \theta_{\text{dep}}(i, j) \\
= \operatorname*{arg\,max}_{\mathbf{x},\mathbf{y},\mathbf{z}} & \; \mathbf{x}^{\top} \boldsymbol{\theta}_{\text{tok}} + \mathbf{y}^{\top} \boldsymbol{\theta}_{\text{ngr}} + \mathbf{z}^{\top} \boldsymbol{\theta}_{\text{dep}} \quad (2)
\end{aligned}$$

where $\boldsymbol{\theta}_{\text{tok}} \triangleq \langle \theta_{\text{tok}}(i) \rangle_{t_i \in T}$ denotes the vector of token scores for all tokens $t_i \in T$ and $\boldsymbol{\theta}_{\text{ngr}}$ and $\boldsymbol{\theta}_{\text{dep}}$ represent vectors of scores for all trigrams and dependency arcs in U and V respectively. The joint objective in (2) is an appealingly straightforward and yet general formulation for the compression task. For instance, the use of standard substructures like n-grams permits scoring of the output sequence configuration $\mathbf{y}$ under probabilistic n-gram language models as in Clarke and Lapata (2008). Similarly, consideration of dependency arcs allows the compressed dependency tree $\mathbf{z}$ to be scored using a rich set of indicator features over dependency labels, part-of-speech tags and even lexical features as in Filippova and Strube (2008).

However, unlike the bag-of-tokens scenario, these output structures cannot be constructed efficiently due to their interdependence. Specifically, we need to maintain the following conditions in order to obtain an interpretable token sequence $\mathbf{y}$:

- Trigram variables y_{ijk} must be non-zero if and only if their corresponding word variables x_i, x_j and x_k are non-zero.
- The non-zero y_{ijk} must form a sentence-like linear ordering, avoiding disjoint structures, cycles and branching.

Similarly, a well-formed dependency tree $\mathbf{z}$ will need to satisfy the following conditions:

- Dependency variables z_{ij} must be non-zero if and only if the corresponding word variables x_i and x_j are.
- The non-zero z_{ij} must form a directed tree with one parent per node, a single root node and no cycles.

2.2 Constrained ILP Formulation

We now discuss an approach to recover exact solutions to (2) under the appropriate structural constraints, thereby yielding globally optimal compressions $\widehat{S} \equiv \langle \mathbf{x}, \mathbf{y}, \mathbf{z} \rangle$ given some input sentence S and model parameters for the scoring functions. For this purpose, we formulate the inference task for joint structural transduction as an integer linear program (ILP)—a type of linear program (LP) in which some or all of the decision variables are restricted to integer values. A number of highly optimized general-purpose solvers exist for solving ILPs thereby making them tractable for sentence-level natural language problems in which the number of variables and constraints is described by a low-order polynomial over the size of the input.

Recent years have seen ILP applied to many structured NLP applications including dependency parsing (Riedel and Clarke, 2006; Martins et al., 2009), text alignment (DeNero and Klein, 2008; Chang et al., 2010; Thadani et al., 2012) and many previous approaches to sentence and document compression (Clarke and Lapata, 2008; Filippova and Strube, 2008; Martins and Smith, 2009; Clarke and Lapata, 2010; Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012).

2.2.1 Basic structural constraints

We start with constraints that define the behavior of terminal tokens. Let y_{*jk}, y_{ij*} and z_{*j} denote indicator variables for the sentence-starting trigram $\langle \textsc{start}, t_j, t_k\rangle$, the sentence-ending trigram $\langle t_i, t_j, \textsc{end}\rangle$ and the root dependency $\langle \textsc{root}, t_j\rangle$ respectively. A valid output sentence will started and terminate with exactly one trigram (perhaps the same); similarly, exactly one word should act as the root of the output dependency tree.

$$\sum_{j,k} y_{*jk} = 1 \tag{3}$$

$$\sum_{i,j} y_{ij*} = 1 \tag{4}$$

$$\sum_{j} z_{*j} = 1 \tag{5}$$

Indicator variables for any substructure, i.e., n-gram or dependency arc, must be kept consistent with the token variables that the substructure is defined over. For instance, we require constraints which specify that tokens can only be active (non-zero) in the solution when, for $1 \leq p \leq n$, there is exactly one active n-gram in the solution which contains this word in position p.[3] Tokens and dependency arcs can similarly be kept consistent by ensuring that a word can only be active when one incoming arc is active.

$$x_l - \sum_{\substack{i,j,k:\\ l\in\{i,j,k\}}} y_{ijk} = 0, \quad \forall t_l \in T \tag{6}$$

$$x_j - \sum_{i} z_{ij} = 0, \quad \forall t_j \in T \tag{7}$$

2.2.2 Flow-based structural constraints

A key challenge for structured transduction models lies in ensuring that output token sequences and dependency trees are well formed. This requires that output structures are fully connected and that cycles are avoided. In order to accomplish this, we introduce additional variables to establish *single-commodity flow* (Magnanti and Wolsey, 1994) between all pairs of tokens, inspired by recent work in dependency parsing (Martins et al., 2009). Linear token ordering is maintained by defining real-valued *adjacency* commodity flow variables γ_{ij}^{adj} which must be non-zero whenever t_j directly follows t_i in an output sentence. Similarly, tree-structured dependencies are enforced using additional *dependency* commodity flow variables γ_{ij}^{dep} which must be non-zero whenever t_j is the dependent of t_i in the output sentence. As with the structural indicators, flow variables γ_{*j}^{adj}, γ_{i*}^{adj}, γ_{*j}^{dep} are also defined for the terminal pseudo-tokens START, END and ROOT respectively.

Each active token in the solution *consumes* one unit of each commodity from the flow variables connected to it. In conjunction with the consistency constraints from equations (6) and (7), this ensures that cycles cannot be present in the flow structure for either commodity.

$$\sum_{i} \gamma_{ij}^{c} - \sum_{k} \gamma_{jk}^{c} = x_j, \quad \forall t_j \in T, \quad \forall c \in \{\text{adj, dep}\} \tag{8}$$

By itself, (8) would simply set all token indicators x_i simultaneously to 0. However, since START and ROOT have no incoming flow variables, the amount of commodity in the respective outgoing flow variables γ_{*j}^{adj} and γ_{*j}^{dep} remains unconstrained. These flow variables therefore provide a point of origin for their respective commodities.

In order for commodity flow to be meaningful, it should be confined to mirroring active structural indicators; for this, we first restrict the amount of commodity in any γ_{ij}^{c} to be non-negative.

$$\gamma_{ij}^{c} \geq 0, \quad \forall t_i, t_j \in T \quad \forall c \in \{\text{adj, dep}\} \tag{9}$$

The adjacency commodity is then linked to the n-grams that would actually establish an adjacency relationship between two tokens, while the dependency commodity is linked to its corresponding dependency arcs. In conjunction with (8–9), these

[3] Note that this does not always hold for n-grams of order $n > 2$ due to the way terminal n-grams featuring START and END are defined. Specifically, in a valid linear ordering of tokens and $\forall r \in 1 \ldots n-2$, there can be no n-grams that feature the last $n-r-1$ tokens in the r'th position or the first $n-r-1$ tokens in the $(n-r+1)$'th position. However, this is easily tackled computationally by assuming that the terminal n-gram replaces these missing n-grams for near-terminal tokens in constraint (6).

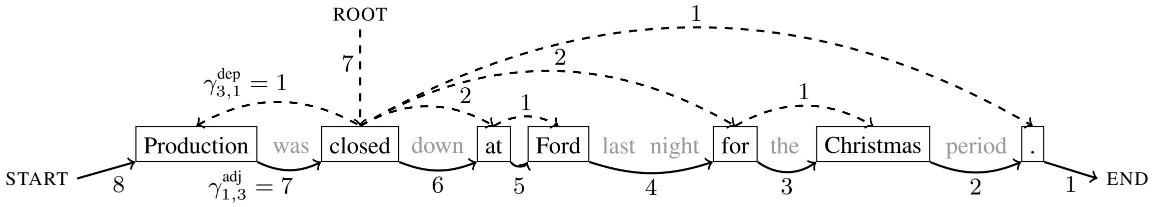

Figure 1: An illustration of commodity values for a valid solution of the program. The adjacency commodity γ^{adj} and dependency commodity γ^{dep} are denoted by solid and dashed lines respectively.

constraints also serve to establish *connectivity* for their respective structures.

$$\gamma^{\text{adj}}_{ij} - C_{\text{max}} \sum_k y_{ijk} \le 0, \quad \forall t_i, t_j \in T \qquad (10)$$

$$\gamma^{\text{adj}}_{jk} - C_{\text{max}} \sum_i y_{ijk} \le 0, \quad \forall t_j, t_k \in T \qquad (11)$$

$$\gamma^{\text{dep}}_{ij} - C_{\text{max}} z_{ij} \le 0, \quad \forall t_i, t_j \in T \qquad (12)$$

where C_{max} is the maximum amount of commodity that the γ_{ij} variables may carry and serves as an upper bound on the number of tokens in the output sentence. Since we use commodity flow to avoid cyclical structure and not to specify spanning arborescences (Martins et al., 2009), C_{max} can simply be set to an arbitrary large value.

2.2.3 Compression rate constraints

The constraints specified above are adequate to enforce structural soundness in an output compression. In addition, compression tasks often involve a restriction on the size of the output sentence. When measured in tokens, this can simply be expressed via constraints over token indicators.

$$\sum_i x_i \ge R_{\text{min}} \qquad (13)$$

$$\sum_i x_i \le R_{\text{max}} \qquad (14)$$

where the compression rate is enforced by restricting the number of output tokens to $[R_{\text{min}}, R_{\text{max}}]$.

2.3 Features

The scoring functions θ that guide inference for a particular compression instance are defined above as linear functions over structure-specific features. We employ the following general classes of features for tokens, trigrams and dependency arcs.

1. **Informativeness**: Good compressions might require specific words or relationships between words to be preserved, highlighted, or perhaps explicitly rejected. This can be expressed through features on token variables that indicate *a priori* salience.[4] For this purpose, we rely on indicator features for part-of-speech (POS) sequences of length up to 3 that surround the token and the POS tag of the token's syntactic governor conjoined with the label. Inspired by McDonald (2006), we also maintain indicator features for stems of verbs (at or adjacent to the token) as these can be useful indications of salience in compression. Finally, we maintain features for whether tokens are negation words, whether they appear within parentheses and if they are part of a capitalized sequence of tokens (an approximation of named entities).
2. **Fluency**: These features are intended to capture how the presence of a given substructure contributes to the overall fluency of a sentence. The n-gram variables are scored with a feature expressing their log-likelihood under an LM. For n-gram variables, we include features that indicate the POS tags and dependency labels corresponding to the tokens it covers. Dependency variable features involve indicators for the governor POS tag conjoined with the dependency direction. In addition, we also use lexical features for prepositions in the governor position of dependency variables in order to indicate whether certain prepositional phrases are likely to be preserved in compressions.
3. **Fidelity**: One might reasonably expect that many substructures in the input sentence will appear unchanged in the output sentence. Therefore, we propose boolean features that indicate that a substructure was seen in the input. Fidelity scores are included for all n-gram variables alongside label-specific fi-

[4] Many compression systems (Clarke and Lapata, 2008; Filippova and Strube, 2008) use a measure based on *tf*idf* which derives from informativeness score of Hori and Furui (2004), but we found this to be less relevant here.

delity scores for dependency arc variables, which can indicate whether particular labels are more or less likely to be dropped.

4. **Pseudo-normalization**: A drawback of using linear models for generation problems is an inability to employ output sentence length normalization in structure scoring. For this purpose, we use the common machine translation (MT) strategy of employing word penalty features. These are essentially word counts whose parameters are intended to balance out the biases in output length which are induced by other features.

Each scale-dependent feature is recorded both absolutely as well as normalized by the length of the input sentence. This is done in order to permit the model to acquire some robustness to variation in input sentence length when learning parameters.

2.4 Learning

In order to leverage a training corpus to recover weight parameters $\mathbf{w}^*$ for the above features that encourage good compressions for unseen data, we rely on the structured perceptron of Collins (2002). A fixed learning rate is used and parameters are averaged to limit overfitting.[5] In our experiments, we observed fairly stable convergence for compression quality over held-out development corpora, with peak performance usually encountered by 10 training epochs.

3 Experiments

In order to evaluate the performance of the structured transduction framework, we ran compression experiments over the newswire (NW) and broadcast news transcription (BN) corpora collected by Clarke and Lapata (2008). Sentences in these datasets are accompanied by gold compressions—one per sentence for NW and three for BN—produced by trained human annotators who were restricted to using word deletion, so paraphrasing and word reordering do not play a role. For this reason, we chose to evaluate the systems using n-gram precision and recall (among other metrics), following Unno et al. (2006) and standard MT evaluations.

We filtered the corpora to eliminate instances with less than 2 and more than 110 tokens and used the same training/development/test splits from Clarke and Lapata (2008), yielding 953/63/603 sentences respectively for the NW corpus and 880/78/404 for the BN corpus. Dependency parses were retrieved using the Stanford parser[6] and ILPs were solved using Gurobi.[7] As a state-of-the-art baseline for these experiments, we used a reimplementation of the LM-based system of Clarke and Lapata (2008), which we henceforth refer to as CL08. This is equivalent to a variant of our proposed model that excludes variables for syntactic structure, uses LM log-likelihood as a feature for trigram variables and a *tf*idf*-based significance score for token variables, and incorporates several targeted syntactic constraints based on grammatical relations derived from RASP (Briscoe et al., 2006) designed to encourage fluent output.

Due to the absence of word reordering in the gold compressions, trigram variables y that were considered in the structured transduction approach were restricted to only those for which tokens appear in the same order as the input as is the case with CL08. Furthermore, in order to reduce computational overhead for potentially-expensive ILPs, we also excluded dependency arc variables which inverted an existing governor-dependent relationship from the input sentence parse.

A recent analysis of approaches to evaluating compression (Napoles et al., 2011b) has shown a strong correlation between the compression rate and human judgments of compression quality, thereby concluding that comparisons of systems which compress at different rates are unreliable. Consequently, all comparisons that we carry out here involve a restriction to a particular compression rate to ensure that observed differences can be interpreted meaningfully.

3.1 Results

Table 1 summarizes the results from compression experiments in which the target compression rate is set to the average gold compression rate for each instance. We observe a significant gain for the joint structured transduction system over the Clarke and Lapata (2008) approach for n-gram F_1. Since n-gram metrics do not distinguish between content words and function words, we also include an evaluation metric that observes the precision, recall and F-measure of nouns and verbs

[5] Given an appropriate loss function, large-margin structured learners such as k-best MIRA (McDonald et al., 2005) can also be used as shown in Clarke and Lapata (2008).

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

[7] `http://www.gurobi.com`

Corpus	System	n-grams $F_1\%$				Content words			Syntactic relations $F_1\%$	
		$n=1$	2	3	4	P%	R%	$F_1\%$	Stanford	RASP
NW	CL08	66.65	53.08	40.35	31.02	73.84	66.41	69.38	51.51	50.21
	Joint ST	**71.91**	**58.67**	**45.84**	**35.62**	**76.82**	**76.74**	**76.33**	**55.02**	50.81
BN	CL08	75.08	61.31	46.76	37.58	80.21	75.32	76.91	60.70	57.27
	Joint ST	**77.82**	**66.39**	**52.81**	**42.52**	80.77	**81.93**	**80.75**	61.38	56.47

Table 1: Experimental results under various quality metrics (see text for descriptions). Systems were restricted to produce compressions that matched their average gold compression rate. Boldfaced entries indicate significant differences ($p < 0.0005$) under the paired t-test and Wilcoxon's signed rank test.

as a proxy for the content in compressed output. From these, we see that the primary contribution of the supervised joint approach is in enhancing the recall of meaning-bearing words.

In addition to the direct measures discussed above, Napoles et al. (2011b) indicate that various other metrics over syntactic relations such as those produced by RASP also correlate significantly with human judgments of compression quality. Compressed sentences were therefore parsed with RASP as well as the Stanford dependency parser and their resulting dependency graphs were compared to those of the gold compressions. These metrics show statistically insignificant differences except in the case of F_1 over Stanford dependencies for the NW corpus.[8]

Comparisons with CL08 do not adequately address the question of whether the performance gain observed is driven by the novel joint inference approach or the general power of discriminative learning. To investigate this, we also studied a variant of the proposed model which eliminates the dependency variables $\mathbf{z}$ and associated commodity flow machinery, thereby bridging the gap between the two systems discussed above. This system, which we refer to as Seq ST, is otherwise trained under similar conditions as Joint ST. Table 2 contains an example of incorrect system output for the three systems under study and illustrates some specific quirks of each, such as the tendency of CL08 to preserve deeply nested noun phrases, the limited ability of Seq ST to identify heads of constituents and the potential for plausible but unusual output parses from Joint ST.

Input	When Los Angeles hosted the Olympics in 1932 , Kurtz competed in high platform diving.
Gold	When Los Angeles hosted the Olympics , Kurtz competed in high diving .
CL08	When Los Angeles hosted Olympics in 1932 , in high platform diving .
Seq ST	When Los Angeles hosted the Olympics , Kurtz competed in high platform
Joint ST	When Los Angeles hosted the Olympics in 1932 , Kurtz competed diving .

Table 2: Examples of erroneous system compressions for a test instance from the NW corpus.

Figure 2 examines the variation of content word $F_1\%$ when the target compression rate is varied for the BN corpus, which contains three reference compressions per instance. Although the gold compressions are often unreachable under low rates, this provides a view into a model's ability to select meaningful words under compression constraints. We observe that the Joint ST model consistently identifies content words more accurately than the sequence-only models despite sharing all token and trigram features with Seq ST.

Figure 3 studies the variation of RASP grammatical relation $F_1\%$ with compression rate as an approximate measure of grammatical robustness. As all three systems track each other fairly closely, the plot conveys the absolute difference of the ST systems from the CL08 baseline, which reveals that Joint ST largely outperforms Seq ST under different compression conditions. We also note that a high compression rate, i.e., minimal compression, is generally favorable to CL08 under the RASP F_1 measure and conjecture that this may be due to the hard syntactic constraints employed by CL08, some of which are defined over RASP relations. At higher compression rates, these constraints largely serve to prevent the loss of meaningful syntactic relationships, e.g., that between a preposition and its prepositional phrase; however, a restrictive compression rate would likely result in all such mutually-constrained components being dropped rather than simultaneously preserved.

[8] Our RASP F_1 results for Clarke and Lapata (2008) in Table 1 outperform their reported results by about 10% (absolute) which may stem from our Gigaword-trained LM or improvements in recent versions of RASP.

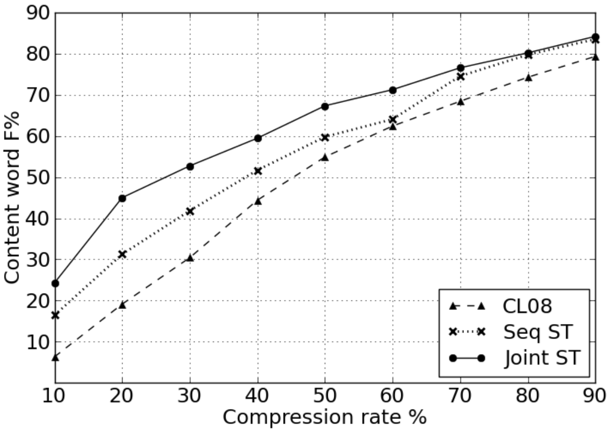

Figure 2: Informativeness of compressions in the BN test corpus indicated by noun and verb $F_1\%$ with respect to gold at different compression rates.

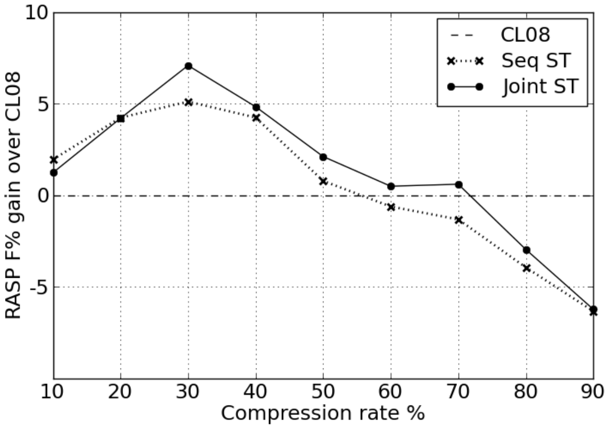

Figure 3: Relative grammaticality of BN test corpus compressions indicated by the absolute difference of RASP relation $F_1\%$ from that of CL08.

4 Related Work

An early notion of compression was proposed by Dras (1997) as reluctant sentence paraphrasing under length constraints. Jing and McKeown (2000) analyzed human-generated summaries and identified a heavy reliance on sentence reduction (Jing, 2000). The extraction by Knight and Marcu (2000) of a dataset of natural compression instances from the Ziff-Davis corpus spurred interest in supervised approaches to the task (Knight and Marcu, 2002; Riezler et al., 2003; Turner and Charniak, 2005; McDonald, 2006; Unno et al., 2006; Galley and McKeown, 2007; Nomoto, 2007). In particular, McDonald (2006) expanded on Knight & Marcu's (2002) transition-based model by using dynamic programming to recover optimal transition sequences, and Clarke and Lapata (2006a) used ILP to replace pairwise transitions with trigrams. Other recent work (Filippova and Strube, 2008; Galanis and Androutsopoulos, 2010) has used dependency trees directly as sentence representations for compression. Another line of research has attempted to broaden the notion of compression beyond mere word deletion (Cohn and Lapata, 2009; Ganitkevitch et al., 2011; Napoles et al., 2011a). Finally, progress on standalone compression tasks has also enabled document summarization techniques that jointly address sentence selection and compression (Daumé and Marcu, 2002; Clarke and Lapata, 2007; Martins and Smith, 2009; Berg-Kirkpatrick et al., 2011; Woodsend and Lapata, 2012), a number of which also rely on ILP-based inference.

Monolingual text-to-text generation research also faces many obstacles common to MT. Recent work in MT decoding has proposed more efficient approaches than ILP to produced text optimally under syntactic and sequential models of language (Rush and Collins, 2011). We are currently exploring similar ideas for compression and other text-to-text generation problems.

5 Conclusion

We have presented a supervised discriminative approach to sentence compression that elegantly accounts for two complementary aspects of sentence structure—token ordering and dependency syntax. Our inference formulation permits rich, linguistically-motivated features that factor over the tokens, n-grams and dependencies of the output. Structural integrity is maintained by linear constraints based on commodity flow, resulting in a flexible integer linear program for the inference task. We demonstrate that this approach leads to significant performance gains over a state-of-the-art baseline compression system without resorting to hand-picked constraints on output content.

Acknowledgments

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D11PC20153. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In *Proceedings of ACL-HLT*, pages 481–490.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the ACL-COLING Interactive Presentation Sessions*.

Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. 2010. Discriminative learning over constrained latent representations. In *Proceedings of HLT-NAACL*, pages 429–437.

James Clarke and Mirella Lapata. 2006a. Constraint-based sentence compression: an integer programming approach. In *Proceedings of ACL-COLING*, pages 144–151.

James Clarke and Mirella Lapata. 2006b. Models for sentence compression: a comparison across domains, training requirements and evaluation measures. In *Proceedings of ACL-COLING*, pages 377–384.

James Clarke and Mirella Lapata. 2007. Modelling compression with discourse constraints. In *Proceedings of EMNLP-CoNLL*, pages 1–11.

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

James Clarke and Mirella Lapata. 2010. Discourse constraints for document compression. *Computational Linguistics*, 36(3):411–441.

Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34(1):637–674, April.

Michael Collins. 2002. Discriminative training methods for hidden Markov models. In *Proceedings of EMNLP*, pages 1–8.

Hal Daumé, III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of ACL*, pages 449–456.

John DeNero and Dan Klein. 2008. The complexity of phrase alignment problems. In *Proceedings of ACL-HLT*, pages 25–28.

Mark Dras. 1997. Reluctant paraphrase: Textual restructuring under an optimisation model. In *Proceedings of PacLing*, pages 98–104.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of INLG*, pages 25–32.

Dimitrios Galanis and Ion Androutsopoulos. 2010. An extractive supervised two-stage method for sentence compression. In *Proceedings of HLT-NAACL*, pages 885–893.

Michel Galley and Kathleen McKeown. 2007. Lexicalized Markov grammars for sentence compression. In *Proceedings of HLT-NAACL*, pages 180–187, April.

Juri Ganitkevitch, Chris Callison-Burch, Courtney Napoles, and Benjamin Van Durme. 2011. Learning sentential paraphrases from bilingual parallel corpora for text-to-text generation. In *Proceedings of EMNLP*, pages 1168–1179.

Chiori Hori and Sadaoki Furui. 2004. Speech summarization: an approach through word extraction and a method for evaluation. *IEICE Transactions on Information and Systems*, E87-D(1):15–25.

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

Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the Conference on Applied Natural Language Processing*, pages 310–315.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization - step one: Sentence compression. In *Proceedings of AAAI*, pages 703–710.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artificial Intelligence*, 139(1):91–107, July.

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

André F. T. Martins and Noah A. Smith. 2009. Summarization with a joint model for sentence extraction and compression. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*, pages 1–9.

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, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL*, pages 91–98.

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

Courtney Napoles, Chris Callison-Burch, Juri Ganitkevitch, and Benjamin Van Durme. 2011a. Paraphrastic sentence compression with a character-based metric: tightening without deletion. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 84–90.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011b. Evaluating sentence compression: pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 91–97.

Tadashi Nomoto. 2007. Discriminative sentence compression with conditional random fields. *Information Processing and Management*, 43(6):1571–1587, November.

Sebastian Riedel and James Clarke. 2006. Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of EMNLP*, pages 129–137.

Stefan Riezler, Tracy H. King, Richard Crouch, and Annie Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar. In *Proceedings of HLT-NAACL*, pages 118–125.

Alexander M. Rush and Michael Collins. 2011. Exact decoding of syntactic translation models through lagrangian relaxation. In *Proceedings of ACL-HLT*, pages 72–82.

Kapil Thadani, Scott Martin, and Michael White. 2012. A joint phrasal and dependency model for paraphrase alignment. In *Proceedings of COLING*, pages 1229–1238.

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

Yuya Unno, Takashi Ninomiya, Yusuke Miyao, and Jun'ichi Tsujii. 2006. Trimming CFG parse trees for sentence compression using machine learning approaches. In *Proceedings of ACL-COLING*, pages 850–857.

Kristian Woodsend and Mirella Lapata. 2012. Multiple aspect summarization using integer linear programming. In *Proceedings of EMNLP*, pages 233–243.