A Comparison of Model Free versus Model Intensive Approaches to Sentence Compression

Tadashi Nomoto
National Institute of Japanese Literature
10-3 Midori Tachikawa
Tokyo 190-0014 Japan
nomoto@acm.org

Abstract

This work introduces a model free approach to sentence compression, which grew out of ideas from Nomoto (2008), and examines how it compares to a state-of-art model intensive approach known as Tree-to-Tree Transducer, or T3 (Cohn and Lapata, 2008). It is found that a model free approach significantly outperforms T3 on the particular data we created from the Internet. We also discuss what might have caused T3's poor performance.

1 Introduction

While there are a few notable exceptions (Hori and Furui, 2004; Yamagata et al., 2006), it would be safe to say that much of prior research on sentence compression has been focusing on what we might call 'model-intensive approaches,' where the goal is to mimic human created compressions as faithfully as possible, using probabilistic and/or machine learning techniques (Knight and Marcu, 2002; Riezler et al., 2003; Turner and Charniak, 2005; McDonald, 2006; Clarke and Lapata, 2006; Cohn and Lapata, 2007; Cohn and Lapata, 2008; Cohn and Lapata, 2009). Because of this, the question has never been raised as to whether a model free approach – where the goal is not to model what humans would produce as compression, but to provide compressions just as useful as those created by human – will offer a viable alternative to model intensive approaches. This is the question we take on in this paper.[1]

[1]One caveat would be in order. By *model free approach*, we mean a particular approach which does not furnish any parameters or weights that one can train on human created compressions. An approach is said to be *model-intensive* if it does. So as far as the present paper is concerned, we might do equally well with a mention of 'model free' ('model-intensive') replaced with 'unsupervised' ('supervised'), or 'non-trainable' ('trainable').

An immediate benefit of the model-free approach is that we could free ourselves from the drudgery of collecting gold standard data from humans, which is costly and time-consuming. Another benefit is intellectual; it opens up an alternative avenue to addressing the problem of sentence compression hitherto under-explored.

Also breaking from the tradition of previous research on sentence compression, we explore the use of naturally occurring data from the Internet as the gold standard. The present work builds on and takes further an approach called 'Generic Sentence Trimmer' (GST) (Nomoto, 2008), demonstrating along the way that it could be adapted for English with relative ease. (GST was originally intended for Japanese.) In addition, to get a perspective on where we stand with this approach, we will look at how it fares against a state-of-the-art model intensive approach known as 'Tree-to-Tree Transducer' (T3) (Cohn, 2008), on the corpus we created.

2 Approach

Nomoto (2008) discusses a two-level model for sentence compression in Japanese termed 'Generic Sentence Trimmer' (GST), which consists of a component dedicated to producing grammatical sentences, and another to reranking sentences in a way consistent with gold standard compressions. For the convenience's sake, we refer to the generation component as 'GST/g' and the ranking part as 'GST/r.' The approach is motivated largely by the desire to make compressed sentences linguistically fluent, and what it does is to retain much of the syntax of the source sentence as it is, in compression, which stands in contrast to Filippova and Strube (2007) and Filippova and Strube (2008), who while working with dependency structure (as we do), took the issue to be something that can be addressed by selecting and reordering constituents that are deemed relevant.

Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 391–399,
Singapore, 6-7 August 2009.

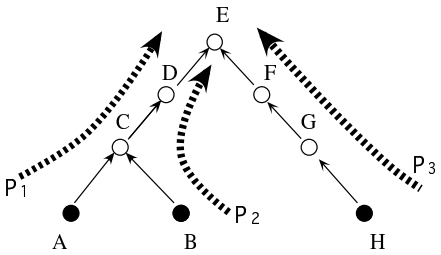

Figure 1: Dependency Structure for 'ABCDEFGH'

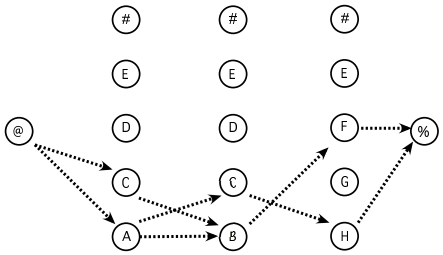

Figure 2: TDP Trellis and POTs.

Getting back to GST, let us consider a sentence,

(1) The bailout plan was likely to depend on private investors to purchase the toxic assets that wiped out the capital of many banks.

Among possible compressions GST/g produces for the sentence are:

(2)

a. The bailout plan was likely to depend on private investors to purchase the toxic assets.

b. The bailout plan was likely to depend on private investors.

c. The bailout plan was likely to depend on investors.

d. The bailout plan was likely.

Notice that they do not differ much from the source sentence (1), except that they get some of the parts chopped off. In the following, we talk about how this could done systematically.

3 Compression with Terminating Dependency Paths

One crucial feature of GST is the notion of *Terminating Dependency Paths* or TDPs, which enables us to factorize a dependency structure into a set of independent fragments. Consider string s = ABCDEFGH with a dependency structure as shown in Figure 1. We begin by locating terminal nodes, i.e., those which have no incoming edges, depicted as filled circles in Figure 1. Next we find a dependency (singly linked) path from each terminal node to the root (labeled E). This would give us three paths p_1 = A-C-D-E, p_2 = B-C-D-E, and p_3 = H-G-F-E (represented by dashed arrows in Figure 1).

Given TDPs, we set out to find a set $\mathcal{T}$ of all suffixes for each TDP, including an empty string, which would look like:

$$\mathcal{T}(p_1) = \{\langle \text{A C D E}\rangle, \langle \text{C D E}\rangle, \langle \text{D E}\rangle, \langle \text{E}\rangle, \langle\rangle\}$$
$$\mathcal{T}(p_2) = \{\langle \text{B C D E}\rangle, \langle \text{C D E}\rangle, \langle \text{D E}\rangle, \langle \text{E}\rangle, \langle\rangle\}$$
$$\mathcal{T}(p_3) = \{\langle \text{G F E}\rangle, \langle \text{F E}\rangle, \langle \text{E}\rangle, \langle\rangle\}$$

Next we combine suffixes, one from each set $\mathcal{T}$, while removing duplicates if any. Combining, for instance, $\langle \text{A C D E}\rangle \in \mathcal{T}(p_1)$, $\langle \text{C D E}\rangle \in \mathcal{T}(p_2)$, and $\langle \text{G F E}\rangle \in \mathcal{T}(p_3)$, would produce {A C D E G F}, which we take to correspond to a string ACDEGF, a short version of s.

As a way of doing this systematically, we put TDPs in a trellis format as in Figure 2, each file representing a TDP, and look for a path across the trellis, which we call 'POT.' It is easy to see that traveling across the trellis (while keeping record of nodes visited), gives you a particular way in which to combine TDPs: thus in Figure 2, we have three POTs, C-B-F, A-C-H, and A-B-F, giving rise to BCDEF, ACDEFGH, and ABCDEF, respectively (where '&' denotes a starting node, '%' an ending node, and '#' an empty string). Note that the POT in effect determines what compression we get.

Take for instance a POT C-B-F. To get to a compression, we first expand C-B-F to get $\{\langle \underline{\text{C}}\ \text{D E}\rangle_1, \langle \underline{\text{B}}\ \text{C D E}\rangle_2, \langle \underline{\text{F}}\ \text{E}\rangle_3\}$ (call it $\mathcal{E}$(C-B-F)). (Note that each TDP is trimmed to start with a node at a corresponding position of the POT.) Next we take a union of TDPs treating them as if they were sets: thus $\bigcup \mathcal{E}(\text{C-B-F}) = \{\text{B C D E F}\}$ = BCDEF.

4 N-Best Search over TDP Trellis

An obvious problem of this approach, however, is that it spawns hundreds of thousands of possible POTs. We would have as many as $5^3 = 125$ of them for the eight-character long string in Figure 1.

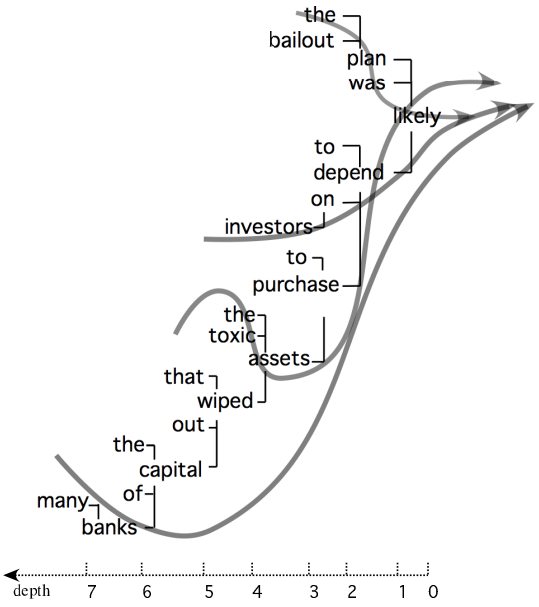

Figure 3: Dependency Structure

What we propose to deal with this problem is to call on a particular ranking scheme to discriminate among candidates we get. Our scheme takes the form of Equation 3 and 4.

$$W(x) = \mathrm{idf}(x) + \exp(-depth(x)) \quad (3)$$

$$S(\mathbf{p}) = \sum_{x_0,\ldots x_n \in \mathcal{E}(\mathbf{p})} W(x_i) \quad (4)$$

$depth(x)$ indicates the distance between x and the root, measured by the number of edges one need to walk across to reach the root from x. Figure 3 shows how the depth is gauged for nodes in a dependency structure. $\mathrm{idf}(x)$ represents the log of the inverse document frequency of x. The equations state that the score S of a POT $\mathbf{p}$ is given as the sum of weights of nodes that comprise $\bigcup \mathcal{E}(\mathbf{p})$.

Despite their being simple, equations 3 and 4 nicely capture our intuition about the way the trimming or compression should work, i.e., that the deeper we go down the tree, or the further away you are from the main clause, the less important information becomes. Putting aside $\mathrm{idf}(x)$ for the moment, we find in Figure 3, $W(assets) > W(capital) > W(banks) > W(many)$. Also depicted in the figure are four TDPs starting with *many*, *the* (preceding *toxic*), *investors*, and *the* (preceding *bailout*).

Finally, we perform a best-first search over the trellis to pick N highest scoring POTs, using Forward DP/Backward A* (Nagata, 1994), with the evaluation function given by Equation 4. We found that the beam search, especially when used with a small width value, does not work as well as the best first search as it tends to produce very short sentences due to its tendency to focus on inner nodes, which generally carry more weights compared to those on the edge. In the experiments described later, we limited the number of candidates to explore at one time to 3,000, to make the search computationally feasible.

Table 1: Drop-me-not rules. A '|' stands for *or*. 'a:b' refers to an element which has both a and b as attributes. Relation names such as *nsubj*, *aux*, *neg*, etc., are from de Marneffe et al. (2006).

R1.	VB	$\Rightarrow$	*nsubj* | *aux* | *neg* | *mark*
R2.	VB	$\Rightarrow$	WDT | WRB
R3.	JJ	$\Rightarrow$	*cop*
R4.	NN	$\Rightarrow$	*det* | *cop*
R5.	NN	$\Rightarrow$	*poss:*WP (='whose')
R6.	$\star$	$\Rightarrow$	*conj* & *cc*

5 'Drop-me-not' Rules

Simply picking a path over the TDP trellis (POT), however, does not warrant the grammaticality of the tree that it generates. Take for instance, a dependency rule, '*likely*←*plan*, *was*, *depend*,' which forms part of the dependency structure for sentence (1). It gives rise to three TDPs, ⟨*plan*, *likely*⟩, ⟨*was*, *likely*⟩, and ⟨*depend*, *likely*⟩. Since we may arbitrarily choose either of the two tokens in each TDP with a complete disregard for a syntagmatic context that each token requires, we may end up with sequences such as '*plan likely*,' '*plan was likely*,' or '*plan likely depend*' (instances of a same token are collapsed into one). This would obviously suggest the need for some device to make the way we pick a path syntagmatically coherent.

The way we respond to the issue is by introducing explicit prohibitions, or 'drop-me-not' rules for POTs to comply with. Some of the major rules are shown in Table 1. A 'drop-me-not' rule (DMN) applies to a local dependency tree consisting of a parent node and its immediate child nodes. The intent of a DMN rule is to prohibit any one of the elements specified on the right hand side of the arrow from falling off in the presence of the head

node; they will be gone only if their head node is.

R1 says that if you have a dependency tree headed by VB with *nsubj*, *aux*, *neg*, or *mark* among its children, they should stay with VB; R2 prohibits against eliminating a WDT or WRB-labeled word in a dependency structure headed by VB; R6 disallows either *cc* or *conj* to drop without accompanying the other, for whatever type the head node assumes.

In Table 2, we find some examples that motivate the kinds of DMN rules we have in Table 1. Note that given the DMNs, the generation of '*was likely depend*,' or '*plan likely depend*' is no longer possible for the sentence in Figure 3.

6 Reranking with CRFs

Pipelining GST/g with CRFs allows us to tap into a host of features found in the sentence that could usefully be exploited toward generating compression, and requires no significant change in the way it is first conceived in Nomoto (2008), in order to make it work for English. It simply involves translating an output by GST/g into the form that allows the use of CRFs; this could be done simply by labeling words included in compression as '1' and those taken out as '0,' which would produce a binary representation of an output generated by GST/g. Given a source sentence $\mathbf{x}$ and a set $G(S)$ of candidate compressions generated by GST/g $-$ represented in binary format $-$ we seek to solve the following,

$$\mathbf{y}^{\star} = \underset{\mathbf{y} \in G(S)}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta}). \tag{5}$$

where $\mathbf{y}^{\star}$ could be found using regular linear-chain CRFs (Lafferty et al., 2001). θ stands for model parameters. In building CRFs, we made use of features representing lexical forms, syntactic categories, dependency relations, TFIDF, whether a given word appears in the title of an article, and the left and right lexical contexts of a word.

7 T3

Cohn and Lapata (2008; 2009) are a recent attempt to bring a machine learning framework known as 'Structured SVM' to bear on sentence compression and could be considered to be among the current state-of-art approaches. Roughly speaking, their approach or what they call 'Tree-to-Tree Transducer' (T3) takes sentence compression to be the problem of classifying the source sentence with its target sentence, where one seeks to find some label $\mathbf{y}$, which represents a compression, for a given source sentence $\mathbf{x}$, that satisfies the following equation,

$$f(\mathbf{x};\mathbf{w}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}}\, F(\mathbf{y},\mathbf{x};\mathbf{w}), \tag{6}$$

and

$$F(\mathbf{y},\mathbf{x};\mathbf{w}) = \langle \mathbf{w}, \Psi(\mathbf{y},\mathbf{x}) \rangle, \tag{7}$$

where $\mathbf{w}$, a vector representing model parameters, is determined in such a way that for a target class $\mathbf{y}$ and a prediction $\mathbf{y}'$, $F(\mathbf{x},\mathbf{y};\mathbf{w}) - F(\mathbf{x},\mathbf{y}';\mathbf{w}) > \Delta(\mathbf{y},\mathbf{y}') - \xi$, $\forall \mathbf{y}' \neq \mathbf{y}$; $\Delta(\mathbf{y},\mathbf{y}')$ represents a loss function and ξ a slack variable (Tsochantaridis et al., 2005). $\Psi(\mathbf{y},\mathbf{x})$ represents a vector of features culled from $\mathbf{y}$ and $\mathbf{x}$, and $\langle \cdot,\cdot \rangle$ a dot product.

For each of the rules used to derive a source sentence, T3 makes a decision on how or whether to transform the rule, with reference to $\langle \cdot,\cdot \rangle$, which takes into account such features as the number of terminals, root category, and lengths of frontiers, which eventually leads to a compression via a chart style dynamic programming.

Table 3: RSS item and its source

R *Two bombings rocked Iraq today, killing at least 80 in attacks at a shrine in Karbala and a police recruiting station in Ramadi.*

S *Baghdad, Jan. 5 – Two new suicide bombings rocked Iraq today, killing at least 80 in an attack at a shrine in the Shiite city of Karbala and a police recruiting station in the Sunni city of Ramadi.*

8 Corpus

Parting ways with previous work on sentence compression, which heavily relied on humans to create gold standard references, this work has a particular focus on using data gathered from RSS feeds, which if successful, could open a door to building gold standard data in large quantities rapidly and with little human effort. The primary objective of the present work is to come up with an approach capable of exploiting naturally occurring data as references for compression. So we are interested

Table 2: Examples. $a \Leftarrow_r b$ means that b stands in an r-relation to a.

rel, *nsubj*	In defying the President, Bill Frist was veering to the political center in a year during which he had artfully courted his party's right wing.	*couted* $\Leftarrow_{rel}$ *during* *veering* $\Leftarrow_{nsubj}$ *Bill Frist*
neg	Isaac B. Weisfuse says that the idea that a pandemic flu will somehow skip the 21st century does not make any sense.	*make* $\Leftarrow_{neg}$ *not*
mark	Prime Minister Ariel Sharon of Israel lashed out at protesters as troops finished clearing all but the last of the 21 Gaza settlements.	*finished* $\Leftarrow_{mark}$ *as*
WDT	The announcement offered few details that would convince protestants that they should resume sharing power with the I.R.A.'s political wing.	*convince* $\Leftarrow_{wdt}$ *that*
WRB	Arbitron, a company best known for its radio ratings, is testing a portable, pager-size device that tracks exposure to media throughout the day, wherever its wearer may go.	*go* $\Leftarrow_{wrb}$ *wherever*
cop	Buildings in a semi-abandoned town just inside Mexico that is a haven for would-be immigrants and smugglers will be leveled.	*haven* $\Leftarrow_{cop}$ *is*
aux, *poss*:WP	Harutoshi Fukui has penned a handful of best sellers whose common themes resonate in a country shedding its pacifism and rearming itself.	*resonate* $\Leftarrow_{poss:\text{WP}}$ *whose* *penned* $\Leftarrow_{aux}$ *has*

Table 4: RSS Corpus from NYTimes.com.

areas	# of items
INTERNATIONAL	2052
NYREGION	1153
NATIONAL	1351
OBITUARIES	541
OPINION	1031
SCIENCE	465
SPORTS	1451
TECHNOLOGY	978
WASHINGTON	1297

in finding out how GST compares with T3 from this particular perspective.

We gathered RSS feeds at NYTimes.com over a period of several months, across different sections, including INTERNATIONAL, NATIONAL, NYREGION, BUSINESS, and so forth, out of which we randomly chose 2,000 items for training data and 116 for testing data. For each RSS summary, we located its potential source sentence in the linked page, using a similarity metric known as SoftTFIDF (Cohen et al., 2003).[2] Table 4 gives a rundown on areas items came from and how many of them we collected for each of these areas.

For the ease of reference, we refer to a corpus of the training and test data combined as 'NYT-RSS,' and let 'NYT-RSS(A)' denote the training part of NYT-RSS, and 'NYT-RSS(B)' the testing part.

[2]SoftTFIDF is a hybrid of the TFIDF scheme and an edit-distance model known as Jaro-Winkler(Cohen et al., 2003).

9 Experiments

We ran the Stanford Parser on NYT-RSS to extract dependency structures for sentences involved, to be used with GST/g (de Marneffe et al., 2006; Klein and Manning, 2003). We manually developed 28 DMN rules out of NYT-RSS(A), some of which are presented in Table 1. An alignment between the source sentence and its corresponding gold standard compression was made by SWA or a standard sequence alignment algorithm by Smith and Waterman (1981). Importantly, we set up GST/g and T3 in such a way that they rely on the same set of dependency analyses and alignments when they are put into operation. We trained T3 on NYT-RSS(A) with default settings except for "–epsilon" and "–delete" options which we turned off, as preliminary runs indicated that their use led to a degraded performance (Cohn, 2008). We also set the loss function as was given in the default settings. We trained both GST/r, and T3 on NYT-RSS(A).

We ran GST/g and GST/g+r, i.e., GST/r pipelined with GST/g, varying the compression rate from 0.4 to 0.7. This involved letting GST/g rank candidate compressions by $S(\mathbf{p})$ and then choosing the first candidate to satisfy a given compression rate, whereas GST/g+r was made to output the highest ranking candidate as measured by $p(\mathbf{y} \mid \mathbf{x}; \theta)$, which meets a particular compression rate. It should be emphasized, however, that in T3, varying compression rate is not something the user has control over; so we accepted whatever output

Table 5: Results on NYT-RSS. '*'-marked figures mean that performance of GST/g is different from that of GST/g+r (on the comparable CompR) at 5% significance level according to t-test. The figures indicate average ratings.

Model	CompR	Intelligibility	Rep.
GST/g+r	0.446	2.836	2.612
GST/g	0.469	3.095	2.569
GST/g+r	0.540	2.957	2.767
GST/g	0.562	3.069	3.026*
GST/g+r	0.632	2.931	2.957
GST/g	0.651	3.060	3.259*
GST/g+r	0.729	3.155	3.345
GST/g	0.743	3.328	3.621*
T3	0.353	1.750	1.586
Gold Std.	0.657	4.776	3.931

T3 generated for a given sentence.

Table 5 shows how GST/g, GST/g+r, and T3 performed on NYT-RSS, along with the gold standard, on a scale of 1 to 5. Ratings were solicited from 4 native speakers of English. 'CompR' indicates compression rate. 'Intelligibility' means how well the compression reads; 'representativeness' how well the compression represents its source sentence. Table 6 presents a guideline for rating, describing what each rating should mean, which was also presented to human judges to facilitate evaluation.

The results in Table 5 indicate a clear superiority of GST/g and GST/g+r over T3, while differences in intelligibility between GST/g and GST/g+r were found not statistically significant. What is intriguing, though, is that GST/g produced performance statistically different in representativeness from GST/g+r at 5% level as marked by the asterisk.

Shown in Table 8 are examples of compression created by GST/g+r, GST/g and T3, together with gold standard compressions and relevant source sentences. One thing worth noting about the examples is that T3 keeps inserting out-of-the-source information into compression, which obviously has done more harm than good to compression.

Table 6: Guideline for Rating

MEANING	EXPLANATION	SCORE
very bad	For intelligibility, it means that the sentence in question is rubbish; no sense can be made out of it. As for representativeness, it means that there is no way in which the compression could be viewed as representing its source.	1
poor	Either the sentence is broken or fails to make sense for the most part, or it is focusing on points of least significance in the source.	2
fair	The sentence can be understood, though with some mental effort; it covers some of the important points in the source sentence.	3
good	The sentence allows easy comprehension; it covers most of important points talked about in the source sentence.	4
excellent	The sentence reads as if it were written by human; it gives a very good idea of what is being discussed in the source sentence.	5

Table 7: Examples from corpora. 'C' stands for reference compression; 'S' source sentence.

NYT-RSS

C *Jeanine F. Pirro said that she would abandon her plans to unseat senator Hillary Rodham Clinton and would instead run for state attorney general .*

S *Jeanine F. Pirro, whose campaign to unseat United States senator Hillary Rodham Clinton was in upheaval almost from the start, said yesterday that she would abandon the race and would instead run for attorney general of New York.*

CLwritten

C *Montserrat, the Caribbean island, is bracing itself for arrests following a fraud investigation by Scotland Yard.*

S *Montserrat, the tiny Caribbean island that once boasted one bank for every 40 inhabitants, is bracing itself this Easter for a spate of arrests following a three-year fraud and corruption investigation by Scotland Yard.*

CLspoken

C *This gives you the science behind the news, with topics explained in detail, from Mad Cow disease to comets.*

S *This page gives you the science behind the news, with hundreds of topics explained in detail, from Mad Cow disease to comets.*

Table 8: form GST/g+r, GST/g, T3, and Gold standard. ('Source' represents a source sentence.)

GST/g+r	The Corporation plans to announce today at the Game Show that it will begin selling the Xbox 360, its new video console , on Nov 22.
GST/g	The Microsoft Corporation plans to announce at the Tokyo Game Show that it will begin selling Xbox 360, new video console , on Nov.
T3	The Microsoft Corporation in New York plans to announce today at the Tokyo Game Show it will begin selling the Xbox 360 , its new video game console, on Nov 22.
Gold	The Microsoft Corporation plans to announce Thursday at the Tokyo Game Show that it will begin selling the Xbox 360 , its new video game console, on Nov. 22.
Source	The Microsoft Corporation plans to announce today at the Tokyo Game Show that it will begin selling the Xbox 360, its new video game console, on Nov 22.
GST/g+r	Scientists may have solved the chemical riddle of why the SARS virus causes such pneumonia and have developed a simple therapy.
GST/g	Scientists may have solved the chemical riddle of why the virus causes such a pneumonia and have developed a simple therapy.
T3	The scientists may solved the chemical riddle of the black river of why the SARS virus causes such a deadly pneumonia.
Gold	Scientists may have solved the riddle of why the SARS virus causes such a deadly pneumonia.
Source	Scientists may have solved the chemical riddle of why the SARS virus causes such a deadly pneumonia and have developed a simple therapy that promises to decrease the extraordinarily high death rate from the disease, according to a report in the issue of the journal nature-medicine that came out this week.
GST/g+r	A flu shot from GlaxoSmithKline was approved by American regulators and the Corporation vaccine plant, shut year because of, moved closer to being opened work to avoid.
GST/g	A flu shot was approved by regulators yesterday and the Chiron Corporation vaccine plant, shut , moved closer to being opened as officials work to avoid shortage.
T3	A flu shot from gaza was the Chiron Corporation's Liverpool vaccine plant, shut last year of a contamination shortage,, but critics suggest he is making it worse.
Gold	The Chiron Corporation's liverpool vaccine plant , shut last year because of contamination, moved closer to being opened as officials work to avoid another shortage.
Source	A flu shot from GlaxoSmithKline was approved by American regulators yesterday and the Chiron Corporation's Liverpool vaccine plant , shut last year because of contamination, moved closer to being opened as officials work to avoid another shortage.

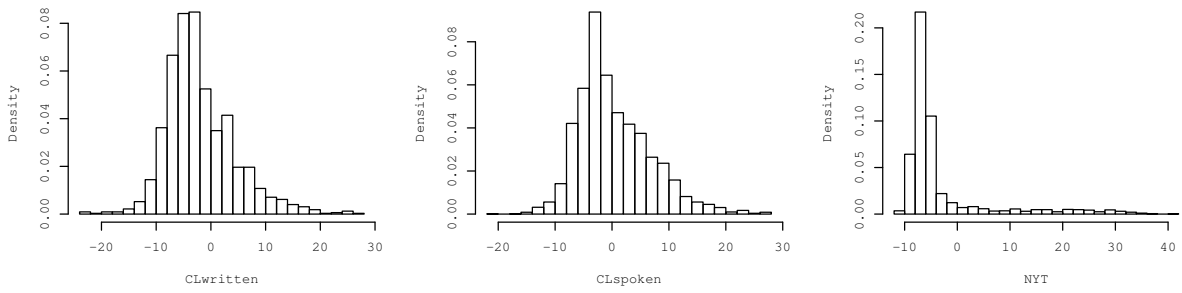

Figure 4: Density distribution of alignment scores. The x-dimension represents the degree of alignment between gold standard compression and its source sentence.

Table 9: Alignment Scores by SWA

NYT-RSS		CLwritten		CLspoken	
-3.061	(2000)	-1.882	(1629)	0.450	(4110)

10 Why T3 fails

It is interesting and worthwhile to ask what caused T3, heavily clad in ideas from the recent machine learning literature, to fail on NYT-RSS, as opposed to the 'CLwritten' and 'CLspoken' corpora on which T3 reportedly prevailed compared to other approaches (Cohn and Lapata, 2009). The CLwritten corpus comes from written sources in the British National Corpus and the American News Text corpus; the CLspoken corpus comes from transcribed broadcast news stories (cf. Table 7).

We argue that there are some important differences between the NYT-RSS corpus and the CLwritten/CLspoken corpora that may have led to T3's poor record with the former corpus.

The CLwritten and CLspoken corpora were created with a specific purpose in mind: namely to assess the compression-by-deletion approach. So their authors had a very good reason to limit gold standard compressions to those that can be arrived at only through deletion; annotators were carefully instructed to create compression by deleting words from the source sentence in a way that preserves the gist of the original sentence. By contrast, NYT-RSS consists of naturally occurring compressions sampled from live feeds on the Internet, where relations between compression and its source sentence are often not as straightforward. For instance, to arrive at a compression in NYT-RSS in Table 7 involves replacing *race* with *her plans to unseat senator Hillary Rodam Clinton*, which is obviously beyond what is possible with the deletion based approach.

In CLwritten and CLspoken, on the other hand, compressions are constructed out of parts that appear *in verbatim* in the original sentence, as Table 7 shows: thus one may get to the compressions by simply crossing off words from the original sentence.

To see whether there is any significant difference among NYT-RSS, CLwritten and CLspoken, we examined how well gold standard compressions are aligned with source sentences on each of the corpora, using SWA. Table 9 shows what we found. Parenthetical numbers represent how many pairs of compression and source are found in each corpus. A larger score means a tighter alignment between gold standard compression and its source sentence: we find in Table 9 that CLspoken has a source sentence more closely aligned with its compression than CLwritten, whose alignments are more closely tied than NYT-RSS's.

Figure 4 (found in the previous page) shows how SWA alignment scores are distributed over each of the corpora. CLwritten and CLspoken have peaks at around 0, with an almost entire mass of scores concentrating in an area close to or above 0. This means that for the most of the cases in either CLwritten or CLspoken, compression is very similar in form to its source. In contrast, NYT-RSS has a heavy concentration of scores in a stretch between -5 and -10, indicating that for the most of time, the overlap between compression and its source is rather modest compared to CLwritten and CLspoken.

So why does T3 fails on NYT-RSS? Because NYT-RSS contains lots of alignments that are only weakly related: in order for T3 to perform well, the training corpus should be made as free of spurious data as possible, so that most of the alignments are rated over and around 0 by SWA. Our concern is that such data may not happen naturally, as the density distribution of NYT-RSS shows, where the majority of alignments are found far below 0, which could raise some questions about the robustness of T3.

11 Conclusions

This paper introduced the model free approach, GST/g, which works by creating compressions only in reference to dependency structure, and looked at how it compares with a model intensive approach T3 on the data gathered from the Internet. It was found that the latter approach appears to crucially rely on the way the corpus is constructed in order for it to work, which may mean a huge compromise.

Interestingly enough, GST/g came out a winner on the particular corpus we used, even outperform-

ing its CRFs harnessed version, GST/g+r in representativeness. This suggests that we might gain more by improving fluency of GST/g than by focusing on its representativeness, which in any case came close to that of human at 70% compression level. The future work should also look at how the present approach fares on CLwritten and CLspoken, for which T3 was found to be effective.

Acknowledgements

The author likes to express gratitude to the reviewers of EMNLP for the time and trouble they took to review the paper. Their efforts are greatly appreciated.

References

James Clarke and Mirella Lapata. 2006. Models for sentence compression: A comparison across domains, training requirements and evaluation measures. In *Proceedings of the 21st COLING and 44th ACL*, pages 377–384, Sydney, Australia, July.

William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. 2003. A comparison of string distance metrics for name-matching tasks. In Subbarao Kambhampati and Craig A. Knoblock, editors, *IIWeb*, pages 73–78.

Trevor Cohn and Mirella Lapata. 2007. Large margin synchronous generation and its application to sentence compression. In *Proceedings of the 2007 EMNLP-CoNLL*, pages 73–82, Prague, Czech Republic, June.

Trevor Cohn and Mirella Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd COLING*, pages 137–144, Manchester, UK, August.

Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. Draft at http://homepages.inf.ed.ac.uk/tcohn/t3/.

Trevor Cohn. 2008. T3: Tree Transducer Toolkit. http://homepages.inf.ed.ac.uk/tcohn/t3/.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC 2006*.

Katja Filippova and Michael Strube. 2007. Generating constituent order in german clauses. In *Proceedings of the 45th ACL*, pages 320–327, Prague, Czech Republic, June.

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

C. 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.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st ACL*, pages 423–430, Sapporo, Japan, July.

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

John Lafferty, Andrew MacCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th ICML-2001*.

Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of the 11th EACL*, pages 297–304.

Masaaki Nagata. 1994. A stochastic japanese morphological analyzer using a forward-dp backward-a* n-best search algorithm. In *Proceedings of COLING-94*.

Tadashi Nomoto. 2008. A generic sentence trimmer with CRFs. In *Proceedings of ACL-08: HLT*, pages 299–307, Columbus, Ohio, June.

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 2003*, pages 118–125, Edmonton.

T. F. Smith and M. S. Waterman. 1981. Identification of common molecular subsequence. *Journal of Molecular Biology*, 147:195–197.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2005. Support vector machine learning for interdependent and structured output spaces. *Journal of Machine Learning Research*, 6:1453–1484.

Jenie Turner and Eugen Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd ACL*, pages 290–297, Ann Arbor, June.

Kiwamu Yamagata, Satoshi Fukutomi, Kazuyuki Takagi, and Kzauhiko Ozeki. 2006. Sentence compression using statistical information about dependency path length. In *Proceedings of TSD 2006 (Lecture Notes in Computer Science, Vol. 4188/2006)*, pages 127–134, Brno, Czech Republic.