Proceedings of HLT-NAACL 2003
Main Papers , pp. 118-125
Edmonton, May-June 2003

Statistical Sentence Condensation using Ambiguity Packing and Stochastic Disambiguation Methods for Lexical-Functional Grammar

Stefan Riezler and **Tracy H. King** and **Richard Crouch** and **Annie Zaenen**
Palo Alto Research Center, 3333 Coyote Hill Rd., Palo Alto, CA 94304
{riezler|thking|crouch|zaenen}@parc.com

Abstract

We present an application of ambiguity packing and stochastic disambiguation techniques for Lexical-Functional Grammars (LFG) to the domain of sentence condensation. Our system incorporates a linguistic parser/generator for LFG, a transfer component for parse reduction operating on packed parse forests, and a maximum-entropy model for stochastic output selection. Furthermore, we propose the use of standard parser evaluation methods for automatically evaluating the summarization quality of sentence condensation systems. An experimental evaluation of summarization quality shows a close correlation between the automatic parse-based evaluation and a manual evaluation of generated strings. Overall summarization quality of the proposed system is state-of-the-art, with guaranteed grammaticality of the system output due to the use of a constraint-based parser/generator.

1 Introduction

Recent work in statistical text summarization has put forward systems that do not merely extract and concatenate sentences, but learn how to generate new sentences from $\langle Summary, Text\rangle$ tuples. Depending on the chosen task, such systems either generate single-sentence "headlines" for multi-sentence text (Witbrock and Mittal, 1999), or they provide a sentence condensation module designed for combination with sentence extraction systems (Knight and Marcu, 2000; Jing, 2000). The challenge for such systems is to guarantee the grammaticality and summarization quality of the system output, i.e. the generated sentences need to be syntactically well-formed and need to retain the most salient information of the original document. For example a sentence extraction system might choose a sentence like:

> *The UNIX operating system, with implementations from Apples to Crays, appears to have the advantage.*

from a document, which could be condensed as:

> *UNIX appears to have the advantage.*

In the approach of Witbrock and Mittal (1999), selection and ordering of summary terms is based on bag-of-words models and n-grams. Such models may well produce summaries that are indicative of the original's content; however, n-gram models seem to be insufficient to guarantee grammatical well-formedness of the system output. To overcome this problem, linguistic parsing and generation systems are used in the sentence condensation approaches of Knight and Marcu (2000) and Jing (2000). In these approaches, decisions about which material to include/delete in the sentence summaries do not rely on relative frequency information on words, but rather on probability models of subtree deletions that are learned from a corpus of parses for sentences and their summaries.

A related area where linguistic parsing systems have been applied successfully is sentence simplification. Grefenstette (1998) presented a sentence reduction method that is based on finite-state technology for linguistic markup and selection, and Carroll et al. (1998) present a sentence simplification system based on linguistic parsing. However, these approaches do not employ statistical learning techniques to disambiguate simplification decisions, but iteratively apply symbolic reduction rules, producing a single output for each sentence.

The goal of our approach is to apply the fine-grained tools for stochastic Lexical-Functional Grammar (LFG) parsing to the task of sentence condensation. The system presented in this paper is conceptualized as a tool that can be used as a standalone system for sentence condensation or simplification, or in combination with sentence extraction for text-summarization beyond the sentence-level. In our system, to produce a condensed version of a sentence, the sentence is first parsed using a broad-coverage LFG grammar for English. The parser produces a set of functional (f)-structures for an ambiguous sentence in a packed format. It presents these to the transfer component in a single packed data structure that represents in one place the substructures shared by several different interpretations. The transfer component operates on these packed representations and modifies the parser output to produce reduced f-structures. The reduced f-structures are then filtered by the generator to determine syntactic well-formedness. A stochastic disambiguator using a maximum entropy model is trained on parsed and manually disambiguated f-structures for pairs of sentences and their condensations. Using the disambiguator, the string generated from the most probable reduced f-structure produced by the transfer system is chosen. In contrast to the approaches mentioned above, our system guarantees the grammaticality of generated strings through the use of a constraint-based generator for LFG which uses a slightly tighter version of the grammar than is used by the parser. As shown in an experimental evaluation, summarization quality of our system is high, due to the combination of linguistically fine-grained analysis tools and expressive stochastic disambiguation models.

A second goal of our approach is to apply the standard evaluation methods for parsing to an automatic evaluation of summarization quality for sentence condensation systems. Instead of deploying costly and non-reusable human evaluation, or using automatic evaluation methods based on word error rate or n-gram match, summarization quality can be evaluated directly and automatically by matching the reduced f-structures that were produced by the system against manually selected f-structures that were produced by parsing a set of manually created condensations. Such an evaluation only requires human labor for the construction and manual structural disambiguation of a reusable gold standard test set. Matching against the test set can be done automatically and rapidly, and is repeatable for development purposes and system comparison. As shown in an experimental evaluation, a close correspondence can be established for rankings produced by the f-structure based automatic evaluation and a manual evaluation of generated strings.

2 Statistical Sentence Condensation in the LFG Framework

In this section, each of the system components will be described in more detail.

2.1 Parsing and Transfer

In this project, a broad-coverage LFG grammar and parser for English was employed (see Riezler et al. (2002)). The parser produces a set of context-free constituent (c-)structures and associated functional (f-)structures for each input sentence, represented in packed form (see Maxwell and Kaplan (1989)). For sentence condensation we are only interested in the predicate-argument structures encoded in f-structures. For example, Fig. 1 shows an f-structure manually selected out of the 40 f-structures for the sentence:

> *A prototype is ready for testing, and Leary hopes to set requirements for a full system by the end of the year.*

The transfer component for the sentence condensation system is based on a component previously used in a machine translation system (see Frank (1999)). It consists of an ordered set of rules that rewrite one f-structure into another. Structures are broken down into flat lists of facts, and rules may add, delete, or change individual facts. Rules may be optional or obligatory. In the case of optional rules, transfer of a single input structure may lead to multiple alternate output structures. The transfer component is designed to operate on packed input from the parser and can also produce packed representations of the condensation alternatives, using methods adapted from parse packing.[1]

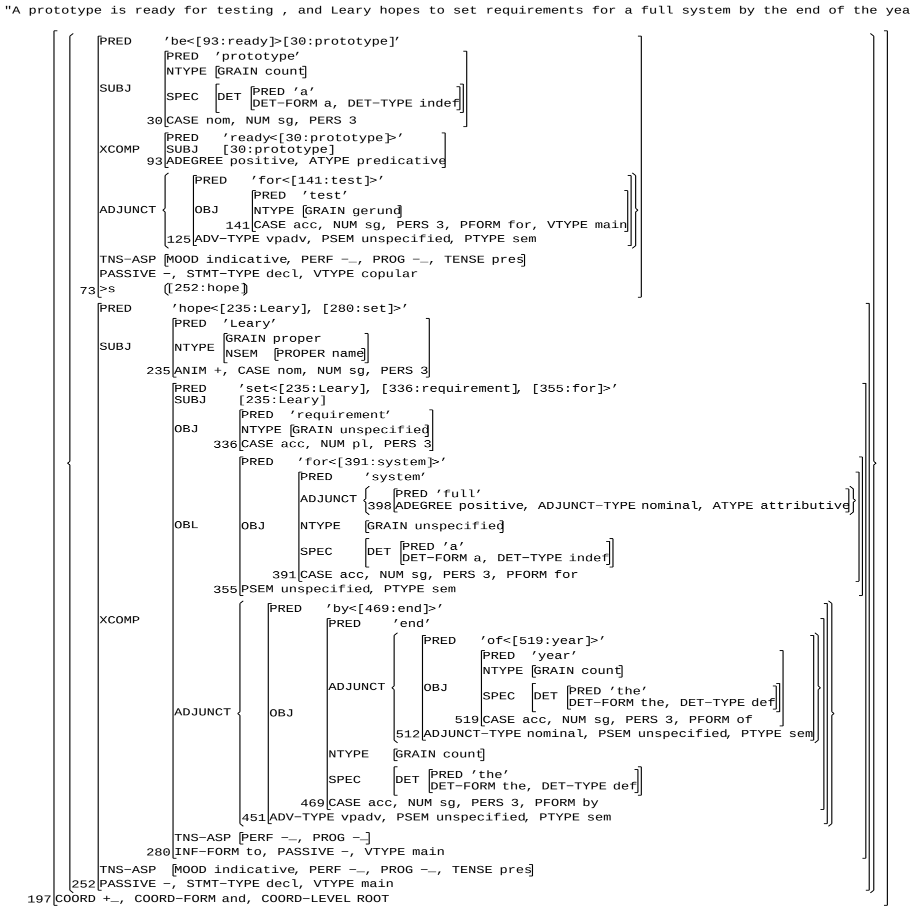
An example rule that (optionally) removes an adjunct is shown below:

```
+adjunct(X,Y), in-set(Z,Y)  ?=>
delete-node(Z,r1), rule-trace(r1,del(Z,X)).
```

This rule eliminates an adjunct, Z, by deleting the fact that Z is contained within the set of adjuncts, Y, associated with the expression X. The + before the `adjunct(X,Y)` fact marks this fact as one that needs to be present for the rule to be applied, but which is left unaltered by the rule application. The `in-set(Z,Y)` fact is deleted. Two new facts are added. `delete-node(Z,r1)` indicates that the structure rooted at node Z is to be deleted, and `rule-trace(r1,del(Z,X))` adds a trace of this rule to an accumulating history of rule applications. This history records the relation of transferred f-structures to the original f-structure and is available for stochastic disambiguation.

Rules used in the sentence condensation transfer system include the optional deletion of all intersective adjuncts (e.g., *He slept in the bed.* can become *He slept.*, but *He did not sleep.* cannot become *He did sleep.* or *He*

[1] The packing feature of the transfer component could not be employed in these experiments since the current interface to the generator and stochastic disambiguation component still requires unpacked representations.



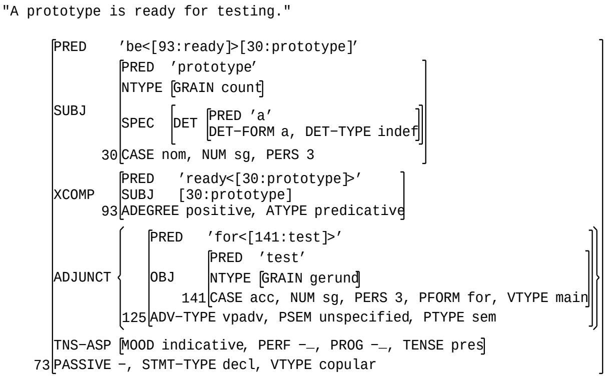
```
"A prototype is ready for testing , and Leary hopes to set requirements for a full system by the end of the yea

        PRED      'be<[93:ready]>[30:prototype]'
                  PRED  'prototype'
                  NTYPE GRAIN count
        SUBJ
                  SPEC  DET  PRED 'a'
                             DET-FORM a, DET-TYPE indef
               30 CASE nom, NUM sg, PERS 3
                  PRED   'ready<[30:prototype]>'
        XCOMP     SUBJ   [30:prototype]
               93 ADEGREE positive, ATYPE predicative
                      PRED   'for<[141:test]>'
                                PRED  'test'
        ADJUNCT       OBJ       NTYPE GRAIN gerund
                            141 CASE acc, NUM sg, PERS 3, PFORM for, VTYPE main
                  125 ADV-TYPE vpadv, PSEM unspecified, PTYPE sem
        TNS-ASP   MOOD indicative, PERF -_, PROG -_, TENSE pres
        PASSIVE -, STMT-TYPE decl, VTYPE copular
     73 >s        ([252:hope])
        PRED       'hope<[235:Leary], [280:set]>'
                   PRED  'Leary'
        SUBJ       NTYPE GRAIN proper
                         NSEM  PROPER name
               235 ANIM +, CASE nom, NUM sg, PERS 3
                   PRED     'set<[235:Leary], [336:requirement], [355:for]>'
                   SUBJ     [235:Leary]
                            PRED  'requirement'
                   OBJ      NTYPE GRAIN unspecified
                        336 CASE acc, NUM pl, PERS 3
                            PRED    'for<[391:system]>'
                                    PRED     'system'
                                    ADJUNCT    PRED 'full'
                                           398 ADEGREE positive, ADJUNCT-TYPE nominal, ATYPE attributive
                   OBL      OBJ     NTYPE    GRAIN unspecified
                                    SPEC     DET PRED 'a'
                                                 DET-FORM a, DET-TYPE indef
                                391 CASE acc, NUM sg, PERS 3, PFORM for
                        355 PSEM unspecified, PTYPE sem
                                PRED    'by<[469:end]>'
                                        PRED     'end'
                                                     PRED    'of<[519:year]>'
                                                             PRED  'year'
                                                             NTYPE GRAIN count
                                        ADJUNCT      OBJ     SPEC  DET PRED 'the'
                                                                       DET-FORM the, DET-TYPE def
        XCOMP                                            519 CASE acc, NUM sg, PERS 3, PFORM of
                                                 512 ADJUNCT-TYPE nominal, PSEM unspecified, PTYPE sem
                   ADJUNCT      OBJ     NTYPE    GRAIN count
                                        SPEC     DET PRED 'the'
                                                     DET-FORM the, DET-TYPE def
                                    469 CASE acc, NUM sg, PERS 3, PFORM by
                            451 ADV-TYPE vpadv, PSEM unspecified, PTYPE sem
                   TNS-ASP  PERF -_, PROG -_
               280 INF-FORM to, PASSIVE -, VTYPE main
        TNS-ASP    MOOD indicative, PERF -_, PROG -_, TENSE pres
    252 PASSIVE -, STMT-TYPE decl, VTYPE main
197 COORD +_, COORD-FORM and, COORD-LEVEL ROOT
```

Figure 1: F-structure for non-condensed sentence.

slept.), the optional deletion of parts of coordinate structures (e.g., *They laughed and giggled.* can become *They giggled.*), and certain simplifications (e.g. *It is clear that the earth is round.* can become *The earth is round.* but *It seems that he is asleep.* cannot become *He is asleep.*). For example, one possible post-transfer output of the sentence in Fig. 1 is shown in Fig. 2.

2.2 Stochastic Selection and Generation

The transfer rules are independent of the grammar and are not constrained to preserve the grammaticality or well-formedness of the reduced f-structures. Some of the reduced structures therefore may not correspond to any English sentence, and these are eliminated from future consideration by using the generator as a filter. The filtering is done by running each transferred structure through the generator to see whether it produces an output string. If it does not, the structure is rejected. For example, for the f-structure in Fig. 1, the transfer system proposed 32 possible reductions. After filtering these structures by generation, 16 reduced f-structures comprising possible



```
"A prototype is ready for testing."

   PRED     'be<[93:ready]>[30:prototype]'
            PRED  'prototype'
            NTYPE GRAIN count
   SUBJ
            SPEC  DET  PRED 'a'
                       DET-FORM a, DET-TYPE indef
         30 CASE nom, NUM sg, PERS 3
            PRED   'ready<[30:prototype]>'
   XCOMP    SUBJ   [30:prototype]
         93 ADEGREE positive, ATYPE predicative
                PRED   'for<[141:test]>'
                          PRED  'test'
   ADJUNCT      OBJ       NTYPE GRAIN gerund
                      141 CASE acc, NUM sg, PERS 3, PFORM for, VTYPE main
            125 ADV-TYPE vpadv, PSEM unspecified, PTYPE sem
   TNS-ASP  MOOD indicative, PERF -_, PROG -_, TENSE pres
73 PASSIVE -, STMT-TYPE decl, VTYPE copular
```

Figure 2: Gold standard f-structure reduction.

condensations of the input sentence survive. The 16 well-formed structures correspond to the following strings that were outputted by the generator (note that a single structure may correspond to more than one string and a given string may correspond to more than one structure):

> *A prototype is ready.*
> *A prototype is ready for testing.*
> *Leary hopes to set requirements for a full system.*
> *A prototype is ready and Leary hopes to set requirements for a full system.*
> *A prototype is ready for testing and Leary hopes to set requirements for a full system.*
> *Leary hopes to set requirements for a full system by the end of the year.*
> *A prototype is ready and Leary hopes to set requirements for a full system by the end of the year.*
> *A prototype is ready for testing and Leary hopes to set requirements for a full system by the end of the year.*

In order to guarantee non-empty output for the overall condensation system, the generation component has to be fault-tolerant in cases where the transfer system operates on a fragmentary parse, or produces non-valid f-structures from valid input f-structures. Robustness techniques currently applied to the generator include insertion and deletion of features in order to match invalid transfer-output to the grammar rules and lexicon. Furthermore, repair mechanisms such as repairing subject-verb agreement from the subject's number value are employed. As a last resort, a fall-back mechanism to the original uncondensed f-structure is used. These techniques guarantee that a non-empty set of reduced f-structures yielding grammatical strings in generation is passed on to the next system component. In case of fragmentary input to the transfer component, grammaticaliy of the output is guaranteed for the separate fragments. In other words, strings generated from a reduced fragmentary f-structure will be as grammatical as the string that was fed into the parsing component.

After filtering by the generator, the remaining f-structures were weighted by the stochastic disambiguation component. Similar to stochastic disambiguation for constraint-based parsing (Johnson et al., 1999; Riezler et al., 2002), an exponential (a.k.a. log-linear or maximum-entropy) probability model on transferred structures is estimated from a set of training data. The data for estimation consists of pairs of original sentences y and gold-standard summarized f-structures s which were manually selected from the transfer output for each sentence. For training data $\{(s_j, y_j)\}_{j=1}^m$ and a set of possible summarized structures $S(y)$ for each sentence y, the objective was to maximize a discriminative criterion, namely the conditional likelihood $L(\boldsymbol{\lambda})$ of a summarized f-structure given the sentence. Optimization of the function shown below was performed using a conjugate gradient optimization routine:

$$L(\boldsymbol{\lambda}) = \log \prod_{j=1}^{m} \frac{e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(s_j)}}{\sum_{s \in S(y_j)} e^{\boldsymbol{\lambda} \cdot \boldsymbol{f}(s)}}.$$

At the core of the exponential probability model is a vector of property-functions $\boldsymbol{f}$ to be weighted by parameters $\boldsymbol{\lambda}$. For the application of sentence condensation, 13,000 property-functions of roughly three categories were used:

- Property-functions indicating attributes, attribute-combinations, or attribute-value pairs for f-structure attributes (≈ 1,000 properties)
- Property-functions indicating co-occurences of verb stems and subcategorization frames (≈ 12,000 properties)
- Property-functions indicating transfer rules used to arrive at the reduced f- structures (≈ 60 properties).

A trained probability model is applied to unseen data by selecting the most probable transferred f-structure, yielding the string generated from the selected structure as the target condensation. The transfered f-structure chosen for our current example is shown in Fig. 3.

```
"A prototype is ready."

  [PRED     'be<[93:ready]>[30:prototype]'                                   ]
  [         [PRED   'prototype'                               ]              ]
  [         [NTYPE  [GRAIN count]                             ]              ]
  [SUBJ     [                                                 ]              ]
  [         [SPEC   [DET  [PRED 'a'                        ]] ]              ]
  [         [       [     [DET-FORM a, DET-TYPE indef      ]] ]              ]
  [       30[CASE nom, NUM sg, PERS 3                         ]              ]
  [         [PRED    'ready<[30:prototype]>'   ]                             ]
  [XCOMP    [SUBJ    [30:prototype]            ]                             ]
  [       93[ADEGREE positive, ATYPE predicative]                            ]
  [TNS-ASP  [MOOD indicative, PERF -_, PROG -_, TENSE pres]                  ]
73[PASSIVE -, STMT-TYPE decl, VTYPE copular                                  ]
```

Figure 3: Transferred f-structure chosen by system.

This structure was produced by the following set of transfer rules, where `var` refers to the indices in the representation of the f-structure:

```
rtrace(r13,keep(var(98),of)),
rtrace(r161,keep(system,var(85))),
rtrace(r1,del(var(91),set,by)),
rtrace(r1,del(var(53),be,for)),
rtrace(r20,equal(var(1),and)),
rtrace(r20,equal(var(2),and)),
rtrace(r2,del(var(1),hope,and)),
rtrace(r22,delb(var(0),and)).
```

These rules delete the adjunct of the first conjunct (*for testing*), the adjunct of the second conjunct (*by the end of the year*), the rest of the second conjunct (*Leary hopes to set requirements for a full system*), and the conjunction itself (*and*).

3 A Method for Automatic Evaluation of Sentence Summarization

Evaluation of quality of sentence condensation systems, and of text summarization and simplification systems in general, has mostly been conducted as intrinsic evaluation by human experts. Recently, Papineni et al.'s (2001) proposal for an automatic evaluation of translation systems by measuring n-gram matches of the system output against reference examples has become popular for evaluation of summarization systems. In addition, an automatic evaluation method based on context-free deletion decisions has been proposed by Jing (2000). However, for summarization systems that employ a linguistic parser as an integral system component, it is possible to employ the standard evaluation techniques for parsing directly to an evaluation of summarization quality. A parsing-based evaluation allows us to measure the semantic aspects of summarization quality in terms of grammatical-functional information provided by deep parsers. Furthermore, human expertise was necessary only for the creation of condensed versions of sentences, and for the manual disambiguation of parses assigned to those sentences. Given such a gold standard, summarization quality of a system can be evaluated automatically and repeatedly by matching the structures of the system output against the gold standard structures. The standard metrics of *precision*, *recall*, and *F-score* from statistical parsing can be used as evaluation metrics for measuring matching quality: *Precision* measures the number of matching structural items in the parses of the system output and the gold standard, out of all structural items in the system output's parse; *recall* measures the number of matches, out of all items in the gold standard's parse. *F-score* balances precision and recall as $(2 \times precision \times recall)/(precision + recall)$.

For the sentence condensation system presented above, the structural items to be matched consist of *relation(predicate, argument)* triples. For example, the gold-standard f-structure of Fig. 2 corresponds to 23 dependency relations, the first 14 of which are shared with the reduced f-structure chosen by the stochastic disambiguation system:

```
tense(be:0, pres),
mood(be:0, indicative),
subj(be:0, prototype:2),
xcomp(be:0, ready:1),
stmt_type(be:0, declarative),
vtype(be:0, copular),
subj(ready:1, prototype:2),
adegree(ready:1, positive),
atype(ready:1, predicative),
det(prototype:2, a:7),
num(prototype:2, sg),
pers(prototype:2, 3),
det_form(a:7, a),
det_type(a:7, indef),
adjunct(be:0, for:12),
obj(for:12, test:14),
adv_type(for:12, vpadv),
psem(for:12, unspecified),
ptype(for:12, semantic),
num(test:14, sg),
pers(test:14, 3),
pform(test:14, for),
vtype(test:14, main).
```

Matching these f-structures against each other corresponds to a precision of 1, recall of .61, and F-score of .76.

The fact that our method does not rely on a comparison of the characteristics of surface strings is a clear advantage. Such comparisons are bad at handling examples which are similar in meaning but differ in word order or vary structurally, such as in passivization or nominalization. Our method handles such examples straightforwardly. Fig. 4 shows two serialization variants of the condensed sentence of Fig. 2. The f-structures for these examples are similar to the f-structure assigned to the gold standard condensation shown in Fig. 2 (except for the relations `ADJUNT-TYPE:parenthetical` versus `ADV-TYPE:vpadv` versus `ADV-TYPE:sadv`). An evaluation of summarization quality that is based on matching f-structures will treat these examples equally, whereas an evaluation based on string matching will yield different quality scores for different serializations.

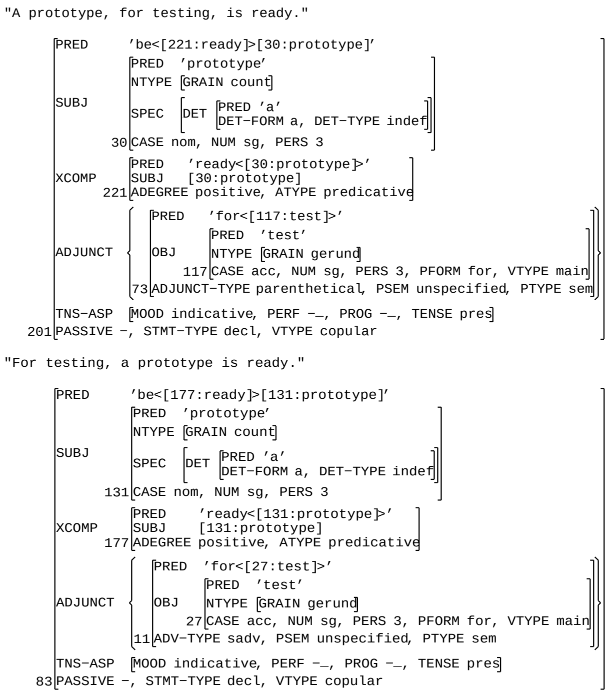

"A prototype, for testing, is ready."
PRED 'be<[221:ready]>[30:prototype]'
SUBJ
PRED 'prototype'
NTYPE GRAIN count
SPEC DET PRED 'a'
DET-FORM a, DET-TYPE indef
30 CASE nom, NUM sg, PERS 3
XCOMP
PRED 'ready<[30:prototype]>'
SUBJ [30:prototype]
221 ADEGREE positive, ATYPE predicative
ADJUNCT
PRED 'for<[117:test]>'
OBJ
PRED 'test'
NTYPE GRAIN gerund
117 CASE acc, NUM sg, PERS 3, PFORM for, VTYPE main
73 ADJUNCT-TYPE parenthetical, PSEM unspecified, PTYPE sem
TNS-ASP MOOD indicative, PERF -_, PROG -_, TENSE pres
201 PASSIVE -, STMT-TYPE decl, VTYPE copular
"For testing, a prototype is ready."
PRED 'be<[177:ready]>[131:prototype]'
SUBJ
PRED 'prototype'
NTYPE GRAIN count
SPEC DET PRED 'a'
DET-FORM a, DET-TYPE indef
131 CASE nom, NUM sg, PERS 3
XCOMP
PRED 'ready<[131:prototype]>'
SUBJ [131:prototype]
177 ADEGREE positive, ATYPE predicative
ADJUNCT
PRED 'for<[27:test]>'
OBJ
PRED 'test'
NTYPE GRAIN gerund
27 CASE acc, NUM sg, PERS 3, PFORM for, VTYPE main
11 ADV-TYPE sadv, PSEM unspecified, PTYPE sem
TNS-ASP MOOD indicative, PERF -_, PROG -_, TENSE pres
83 PASSIVE -, STMT-TYPE decl, VTYPE copular

Figure 4: F-structure for word-order variants of gold standard condensation.

In the next section, we present experimental results of an automatic evaluation of the sentence condensation system described above. These results show a close correspondence between automatically produced evaluation results and human judgments on the quality of generated condensed strings.

4 Experimental Evaluation

The sentences and condensations we used are taken from data for the experiments of Knight and Marcu (2000), which were provided to us by Daniel Marcu. These data consist of pairs of sentences and their condensed versions that have been extracted from computer-news articles and abstracts of the Ziff-Davis corpus. Out of these data, we parsed and manually disambiguated 500 sentence pairs. These included a set of 32 sentence pairs that were used for testing purposes in Knight and Marcu (2000). In order to control for the small corpus size of this test set, we randomly extracted an additional 32 sentence pairs from the 500 parsed and disambiguated examples as a second test set. The rest of the 436 randomly selected sentence pairs were used to create training data. For the purpose of discriminative training, a gold-standard of transferred f-structures was created from the transfer output and the manually selected f-structures for the condensed strings. This was done automatically by selecting for each example the transferred f-structure that best matched the f-structure annotated for the condensed string.

In the automatic evaluation of f-structure match, three different system variants were compared. Firstly, randomly chosen transferred f-structures were matched against the manually selected f-structures for the manually created condensations. This evaluation constitutes a lower bound on the F-score against the given gold standard. Secondly, matching results for transferred f-structures yielding the maximal F-score against the gold standard were recorded, giving an upper bound for the system. Thirdly, the performance of the stochastic model within the range of the lower bound and upper bound was measured by recording the F-score for the f-structure that received highest probability according to the learned distribution on transferred structures.

In order to make our results comparable to the results of Knight and Marcu (2000) and also to investigate the correspondence between the automatic evaluation and human judgments, a manual evaluation of the strings generated by these system variants was conducted. Two human judges were presented with the uncondensed surface string and five condensed strings that were displayed in random order for each test example. The five condensed strings presented to the human judges contained (1) strings generated from three randomly selected f-structures, (2) the strings generated from the f-structures which were selected by the stochastic model, and (3) the manually created gold-standard condensations extracted from the Ziff-Davis abstracts. The judges were asked to judge summarization quality on a scale of increasing quality from 1 to 5 by assessing how well the generated strings retained the most salient information of the original uncondensed sentences. Grammaticality of the system output is optimal and not reported separately. Results for both evaluations are reported for two test corpora of 32 examples each. *Testset I* contains the sentences and condensations used to evaluate the system described in Knight and Marcu (2000). *Testset II* consists of another randomly extracted 32 sentence pairs from the same domain, prepared in the same way.

Fig. 5 shows evaluation results for a sentence condensation run that uses manually selected f-structures for the original sentences as input to the transfer component. These results demonstrate how the condenstation system performs under the optimal circumstances when the parse chosen as input is the best available. Fig. 6 applies the same evaluation data and metrics to a sentence condensation experiment that performs transfer from packed f-structures, i.e. transfer is performed on all parses for an ambiguous sentence instead of on a single manually selected parse. Alternatively, a single input parse could be selected by stochastic models such as the one described in Riezler et al. (2002). A separate phase of parse disambiguation, and perhaps the effects of any errors that this might introduce, can be avoided by transferring from all parses for an ambiguous sentence. This approach is computationally feasible, however, only if condensation can be carried all the way through without unpacking. Our technology is not yet able to do this (in particular, as mentioned earlier, we have not yet implemented a method for stochastic disambiguation on packed f-structures). However, we conducted a preliminary assessment of this possibility by unpacking and enumerating the transferred f-structures. For many sentences this resulted in more candidates than we could operate on in the available time and space, and in those cases we arbitrarily set a cut-off on the number of transferred f-structures we considered. Since transferred f-structures are produced according to the number of rules applied to transfer them, in this setup the transfer system produces smaller f-structures first, and cuts off less condensed output. The result of this experiment, shown in Fig. 6, thus provides a conservative estimate on the quality of the condensations we might achieve with a full-packing implementation.

In Figs. 5 and 6, the first row shows F-scores for a random selection, the system selection, and the best possible selection from the transfer output against the gold standard. The second rows show summarization quality scores for generations from a random selection and the system selection, and for the human-written condensation. The third rows report compression ratios. As can

testset I	lower bound	system selection	upper bound
F-score	58%	67.3%	77.2 %
sum-quality	2.0	3.5	4.4
compr.	50.2%	60.4%	54.9%

testset II	lower bound	system selection	upper bound
F-score	59%	65.4%	83.3%
sum-quality	2.1	3.4	4.6
compr.	52.7%	65.9%	56.8%

Figure 5: Sentence condensation from manually selected *f*-structure for original uncondensed sentences.

testset I	lower bound	system selection	upper bound
F-score	55.2%	63.0%	72.0%
sum-quality	2.1	3.4	4.4
compres.	46.5%	61.6%	54.9%

testset II	lower bound	system selection	upper bound
F-score	54%	59.7%	76.0 %
sum-quality	1.9	3.3	4.6
compres.	50.9%	60.0%	56.8%

Figure 6: Sentence condensation from packed *f*-structures for original uncondensed sentences.

be seen from these tables, the ranking of system variants produced by the automatic and manual evaluation confirm a close correlation between the automatic evaluation and human judgments. A comparison of evaluation results across colums, i.e. across selection variants, shows that a stochastic selection of transferred *f*-structures is indeed important. Even if all *f*-structures are transferred from the same linguistically rich source, and all generated strings are grammatical, a reduction in error rate of around 50% relative to the upper bound can be achieved by stochastic selection. In contrast, a comparison between transfer runs with and without perfect disambiguation of the original string shows a decrease of about 5% in F-score, and of only .1 points for summarization quality when transferring from packed parses instead of from the manually selected parse. This shows that it is more important to learn what a good transferred *f*-structure looks like than to have a perfect *f*-structure to transfer from. The compression rates associated with the systems that used stochastic selection is around 60%, which is acceptable, but not as aggressive as human-written condensations. Note that in our current implementation, in some cases the transfer component was unable to operate on the packed representation. In those cases a parse was chosen at random as a conservative estimate of transfer from all parses. This fall-back mechanism explains the drop in F-score for the upper bound in comparing Figs. 5 and 6.

5 Conclusion

We presented an approach to sentence condensation that employs linguistically rich LFG grammars in a parsing/generation-based stochastic sentence condensation system. Fine-grained dependency structures are output by the parser, then modified by a highly expressive transfer system, and filtered by a constraint-based generator. Stochastic selection of generation-filtered reduced structures uses a powerful Maximum-Entropy model. As shown in an experimental evaluation, summarization quality of the system output is state-of-the-art, and grammaticality of condensed strings is guaranteed. Robustness techniques for parsing and generation guarantee that the system produces non-empty output for unseen input.

Overall, the summarization quality achieved by our system is similar to the results reported in Knight and Marcu (2000). This might seem disappointing considering the more complex machinery employed in our approach. It has to be noted that these results are partially due to the somewhat artificial nature of the data that were used in the experiments of Knight and Marcu (2000) and therefore in our experiments: The human-written condensations in the data set extracted from the Ziff-Davis corpus show the same word order as the original sentences and do not exhibit any structural modification that are common in human-written summaries. For example, humans tend to make use of structural modifications such as nominalization and verb alternations such as active/passive or transitive/intransitive alternations in condensation. Such alternations can easily be expressed in our transfer-based approach, whereas they impose severe problems to approaches that operate only on phrase structure trees. In the given test set, however, the condensation task restricted to the operation of deletion. A creation of additional condensations for the original sentences other than the condensed versions extracted from the human-written abstracts would provide a more diverse test set, and furthermore make it possible to match each system output against any number of independent human-written condensations of the same original sentence. This idea of computing matching scores to multiple reference examples was proposed by Alshawi et al. (1998), and later by Papineni et al. (2001) for evaluation of machine translation systems. Similar to these proposals, an evaluation of condensation quality could consider multiple reference condensations and record the matching score against the most similar example.

Another desideratum for future work is to carry condensation all the way through without unpacking at any stage. Work on employing packing techniques not only for parsing and transfer, but also for generation and stochastic selection is currently underway (see Geman and Johnson (2002)). This will eventually lead to a system whose components work on packed representations of all or n-best solutions, but completely avoid costly unpacking of representations.

References

Hiyan Alshawi, Srinivas Bangalore, and Shona Douglas. 1998. Automatic acquisition of hierarchical transduction models for machine translation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL'98)*, Montreal, Quebec, Canada.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of the AAAI Workshop on Integrating Artificial Intelligence and Assistive Technology*, Madison, WI.

Anette Frank. 1999. From parallel grammar development towards machine translation. In *Proceedings of the MT Summit VII. MT in the Great Translation Era*, pages 134–142. Kent Ridge Digital Labs, Singapore.

Stuart Geman and Mark Johnson. 2002. Dynamic programming for parsing and estimation of stochastic unification-based grammars. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA.

Gregory Grefenstette. 1998. Producing intelligent telegraphic text reduction to provide an audio scanning service for the blind. In *Proceedings of the AAAI Spring Workshop on Intelligent Text Summarization*, Stanford, CA.

Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP'00)*, Seattle, WA.

Mark Johnson, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99)*, College Park, MD.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization—step one: Sentence compression. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX.

John Maxwell and Ronald M. Kaplan. 1989. An overview of disjunctive constraint satisfaction. In *Proceedings of the International Workshop on Parsing Technologies*, Pittsburgh, PA.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. Bleu: a method for automatic evaluation of machine translation. Technical Report IBM Research Division Technical Report, RC22176 (W0190-022), Yorktown Heights, N.Y.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA.

Michael J. Witbrock and Vibhu O. Mittal. 1999. Ultra-summarization: A statistical approach to generating highly condensed non-extractive summaries. In *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA.