Automatic Extraction of Briefing Templates

Dipanjan Das **Mohit Kumar** **Alexander I. Rudnicky**
Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
{dipanjan, mohitkum, air}@cs.cmu.edu

Abstract

An approach to solving the problem of automatic briefing generation from non-textual events can be segmenting the task into two major steps, namely, extraction of briefing templates and learning aggregators that collate information from events and automatically fill up the templates. In this paper, we describe two novel unsupervised approaches for extracting briefing templates from human written reports. Since the problem is non-standard, we define our own criteria for evaluating the approaches and demonstrate that both approaches are effective in extracting domain relevant templates with promising accuracies.

1 Introduction

Automated briefing generation from non-textual events is an unsolved problem that currently lacks a standard approach in the NLP community. Broadly, it intersects the problem of language generation from structured data and summarization. The problem is relevant in several domains where the user has to repeatedly write reports based on events in the domain, for example, weather reports (Reiter et al., 2005), medical reports (Elhadad et al., 2005), weekly class project reports (Kumar et al., 2007) and so forth. On observing the data from these domains, we notice a *templatized* nature of report items. Examples (1)-(3) demonstrate equivalents in a particular domain (Reiter et al., 2005).

(1) [A warm front] from [Iceland] to [northern Scotland] will move [SE] across [the northern North Sea] [today and tomorrow]

(2) [A warm front] from [Iceland] to [the Faeroes] will move [ENE] across [the Norwegian Sea] [this evening]

(3) [A ridge] from [the British Isles] to [Iceland] will move [NE] across [the North Sea] [today]

In each sentence, the phrases in square brackets at the same relative positions form the slots that take up different values at different occasions. The corresponding template is shown in (4) with slots containing their respective domain entity types. Instantiations of (4) may produce (1)-(3) and similar sentences. This kind of sentence structure motivates an approach of segmenting the problem of closed domain summarization into two major steps of automatic template extraction and learning aggregators, which are pattern detectors that assimilate information from the events, to populate these templates.

(4) [PRESSURE ENTITY] from [LOCATION] to [LOCATION] will move [DIRECTION] across [LOCATION] [TIME]

In the current work we address the first problem of automatically extracting domain templates from human written reports. We take a two-step approach to the problem; first, we cluster report sentences based on similarity and second, we extract template(s) corresponding to each cluster by aligning the instances in the cluster. We experimented with two independent, arguably complementary techniques for clustering and aligning – a predicate argument based approach that extracts more general templates containing one predicate and a ROUGE (Lin, 2004) based

approach that can extract templates containing multiple verbs. As we will see below, both approaches show promise.

2 Related Work

There has been instances of template based summarization in popular Information Extraction (IE) evaluations like MUC (Marsh & Perzanowski, 1998; Onyshkevych, 1994) and ACE (ACE, 2007) where hand engineered slots were to be filled for events in text; but the focus lay on template filling rather than their creation. (Riloff, 1996) describes an interesting work on the generation of extraction patterns from untagged text, but the analysis is syntactic and the patterns do not resemble the templates that we aim to extract. (Yangarber et al., 2000) describe another system called ExDisco, that extracts event patterns from un-annotated text starting from seed patterns. Once again, the text analysis is not deep and the patterns extracted are not sentence surface forms.

(Collier, 1998) proposed automatic domain template extraction for IE purposes where MUC type templates for particular types of events were constructed. The method relies on the idea from (Luhn, 1958) where statistically significant words of a corpus were extracted. Based on these words, sentences containing them were chosen and aligned using subject-object-verb patterns. However, this method did not look at arbitrary syntactic patterns.

(Filatova et al., 2006) improved the paradigm by looking at the most frequent verbs occurring in a corpus and aligning subtrees containing the verb, by using the syntactic parses as a similarity metric. However, long distance dependencies of verbs with constituents were not looked at and deep semantic analysis was not performed on the sentences to find out similar verb subcategorization frames. In contrast, in our predicate argument based approach we look into deeper semantic structures, and align sentences not only based on similar syntactic parses, but also based on the constituents' roles with respect to the main predicate. Also, they relied on typical Named Entities (NEs) like location, organization, person etc. and included another entity that they termed as NUMBER. However, for specific domains like weather forecasts, medical reports or student reports, more varied domain entities form slots in templates, as we observe in our data; hence, existence of a module handling domain specific entities become essential for such a task. (Surdeanu et al., 2003) identify arguments for predicates in a sentence and emphasize how semantic role information may assist in IE related tasks, but their primary focus remained on the extraction of PropBank (Kingsbury et al., 2002) type semantic roles.

To our knowledge, the ROUGE metric has not been used for automatic extraction of templates.

3 The Data

3.1 Data Description

Since our focus is on creating summary items from events or structured data rather than from text, we used a corpus from the domain of weather forecasts (Reiter et al., 2005). This is a freely available parallel corpus[1] consisting of weather data and human written forecasts describing them. The dataset showed regularity in sentence structure and belonged to a closed domain, making the variations in surface forms more constrained than completely free text. After sentence segmentation we arrived at a set of 3262 sentences. From this set, we selected 3000 for template extraction and kept aside 262 sentences for testing.

3.2 Preprocessing

For semantic analysis, we used the ASSERT toolkit (Pradhan et al., 2004) that produces shallow semantic parses using the PropBank conventions. As a by product, it also produces syntactic parses of sentences, using the Charniak parser (Charniak, 2001). For each sentence, we maintained a part-of-speech tagged (leaves of the parse tree), parsed, baseNP[2] tagged and semantic role tagged version. The baseNPs were retrieved by pruning the parse trees and not by using a separate NP chunker. The reason for having a baseNP tagged corpus will become clear as we go into the detail of our template extraction techniques. Figure 1 shows a typical output from the Charniak parser and Figure 2 shows the same tree with nodes under the baseNPs pruned.

We identified the need to have a domain entity tagger for matching constituents in the sentences.

[1] http://www.csd.abdn.ac.uk/research/sumtime/

[2] A baseNP is a noun-phrase with no internal noun-phrase

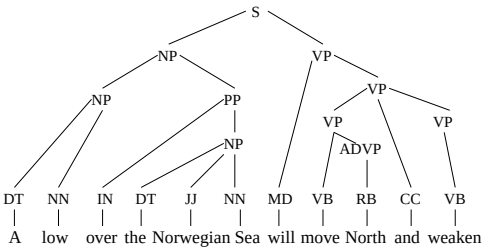

Figure 1: Parse tree for a sentence in the data.

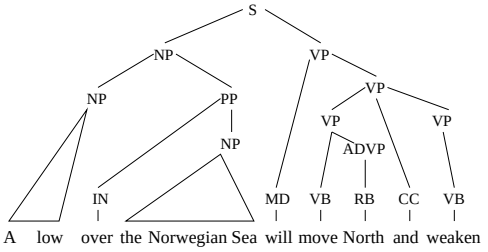

Figure 2: Pruned parse tree for a sentence in the corpus

Any tagger for named entities was not suitable for weather forecasts since unique constituent types assumed significance unlike newswire data. Since the development of such a tagger was beyond the scope of the present work, we developed a module that took baseNP tagged sentences as input and produced tags across words and baseNPs that were domain entities. The development of such a module by hand was easy because of a limited vocabulary ($<$ 1000 words) of the data and the closed set nature of most entity types (e.g the *direction* entity could take up a finite set of values). From inspection, thirteen distinct entity types were recognized in the domain. Figure 3 shows an example output from the entity recognizer with the sentence from Figure 2 as input.

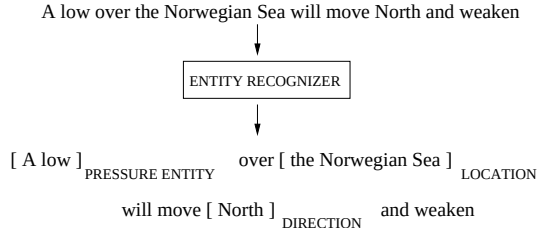

Figure 3: Example output of the entity recognizer

We now provide a detailed description of our clustering and template extraction algorithms.

4 Approach and Experiments

We adopted two parallel approaches. First, we investigated a predicate-argument based approach where we consider the set of all propositions in our dataset, and cluster them based on their verb subcategorization frame. Second, we used ROUGE, a summarization evaluation metric that is generally used to compare machine generated and human written summaries. We uniquely used this metric for clustering similar summary items, after abstracting the surface forms to a representation that facilitates comparison of a pair of sentences. The following subsections detail both the techniques.

4.1 A Predicate-Argument Based Approach

Analysis of predicate-argument structures seemed appropriate for template extraction for a few reasons: Firstly, complicated sentences with multiple verbs are broken down into propositions by a semantic role labeler. The propositions[3] are better generalizable units than whole sentences across a corpus. Secondly, long distance dependencies of constituents with a particular verb, are captured well by a semantic role labeler. Finally, if verbs are considered to be the center of events, then groups of sentences with the same semantic role sequences seemed to form clusters conveying similar meaning. We explain the complete algorithm for template extraction in the following subsections.

(5) `[ARG0 A low over the Norwegian Sea] [AGM-MOD will] [TARGET move ] [ARGM-DIR North ] and weaken`

(6) `[ARG0 A high pressure area ] [AGM-MOD will ] [TARGET move] [ARGM-DIR southwestwards] and build on Sunday.`

4.1.1 Verb based clustering

We performed a verb based clustering as the first step. Instead of considering a unique set of verbs, we considered related verbs as a single verb type. The relatedness of verbs was derived from Wordnet (Fellbaum, 1998), by merging verbs that appear in the same synset. This kind of clustering is not

[3] sentence fragments with one verb

ideal in a corpus containing a huge variation in event streams, like newswire. However, the results were good for the weather domain where the number of verbs used is limited. The grouping procedure resulted in a set of 82 clusters with 6632 propositions.

4.1.2 Matching Role Sequences

Each verb cluster was considered next. Instead of finding structural similarities of the propositions in one go, we first considered the semantic role sequences for each proposition. We searched for propositions that had exactly similar role sequences and grouped them together. To give an example, both sentences 5 and 6 have the matching role sequence ARG0–ARGM-MOD–TARGET–ARGM-DIR. The intuition behind such clustering is straightforward. Propositions with a matching verb type with the same set of roles arranged in a similar fashion would convey similar meaning. We observed that this was indeed true for sentences tagged with correct semantic role labels.

Instead of considering matching role sequences for a set of propositions, we could as well have considered matching bag of roles. However, for the present corpus, we decided to use strict role sequence instead because of the sentences' rigid structure and absence of any passive sentences. This subclustering step resulted in smaller clusters, and many of them contained a single proposition. We threw out these clusters on the assumption that the human summarizers did not necessarily have a template in mind while writing those summary items. As a result, many verb types were eliminated and only 33 verb-type clusters containing several subclusters each were produced.

4.1.3 Looking inside Roles

Groups of propositions with the same verb-type and semantic role sequences were considered in this step. For each group, we looked at individual semantic roles to find out similarity between them. We decided at first to look at syntactic parse tree similarities between constituents. However, there is a need to decide at what level of abstraction should one consider matching the parse trees. After considerable speculation, we decided on pruning the constituents' parse trees till the level of baseNPs and then match the resulting tag sequences.

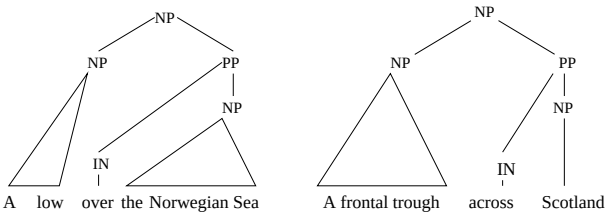

Figure 4: Matching ARG0s for two propositions

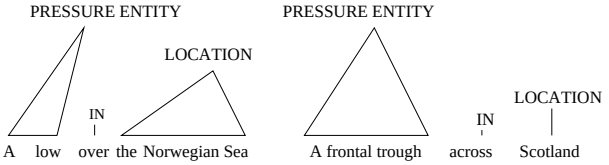

Figure 5: Abstracted tag sequences for two constituents

The parses with pruned trees from the preprocessing steps provide the necessary information for constituent matching. Figure 4 shows matching syntactic trees for two ARG0s from two propositions of a cluster. It is at this step that we use the domain entity tags to abstract away the constituents' syntactic tags. Figure 5 shows the constituents of Figure 4 with the tree structure reduced to tag sequences and domain entity types replacing the tags whenever necessary.

This abstraction step produces a number of unique domain entity augmented tag sequences for a particular semantic role. As a final step of template generation, we concatenate these abstracted constituent types for all the semantic roles in the given group.

To focus on template-like structures we only consider tag sequences that occur twice or more in the group.

The templates produced at the end of this step are essentially tag sequences interspersed with domain entities. In our definition of templates, the slots are the entity types and the fixed parts are constituted by word(s) used by the human experts for a particular tag sequence. Figure 6 shows some example templates. The upper case words in the figure correspond to the domain entities identified by the entity tagger and they form the slots in the templates. A total of 209 templates were produced.

PRESSURE_ENTITY expected over LOCATION by_0.5/on_0.5 DAY
PRESSURE_ENTITY to DIRECTION of LOCATION will drift slowly
WAVE will run_0.5/move_0.5 DIRECTION then DIRECTION
Associated PRESSURE_ENTITY will move DIRECTION across LOCATION TIME

Figure 6: Example Templates. Upper case tokens correspond to slots. For fixed parts, when there is a choice between words, the probability of the occurrence of words in that particular syntactic structure are tagged alongside.

4.2 A ROUGE Based Approach

ROUGE (Lin, 2004) is the standard automatic evaluation metric in the Summarization community. It is derived from the BLEU (Papineni et al., 2001) score which is the evaluation metric used in the Machine Translation community. The underlying idea in the metric is comparing the candidate and the reference sentences (or summaries) based on their token co-occurrence statistics. For example, a unigram based measure would compare the vocabulary overlap between the candidate and reference sentences. Thus, intuitively, we may use the ROUGE score as a measure for clustering the sentences. Amongst the various ROUGE statistics, the most appealing is Weighted Longest Common Subsequence(WLCS). WLCS favors contiguous LCS which corresponds to the intuition of finding the common template. We experimented with other ROUGE statistics but we got better and easily interpretable results using WLCS and so we chose it as the final metric. In all the approaches the data was first preprocessed (baseNP and NE tagged) as described in the previous subsection. In the following subsections, we describe the various clustering techniques that we tried using the ROUGE score followed by the alignment technique.

4.2.1 Clustering

Unsupervised Clustering: As the ROUGE score defines a distance metric, we can use this score for doing unsupervised clustering. We tried hierarchical clustering approaches but did not obtain good clusters, evaluated empirically. In empirical evaluation, we manually looked at the output clusters and made a judgement call whether the candidate clusters are reasonably coherent and potentially correspond to templates. The reason for the poor performance of the approach was the classical parameter estimation problem of determining a priori the number of clusters. We could not find an elegant solution for the problem without losing the motivation of an automated approach.

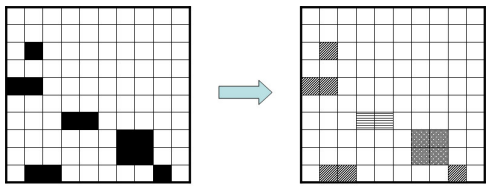

Figure 7: Deterministic clustering based on Graph connectivity. In the figure the squares with the same pattern belong to the same cluster.

Non-parametric Unsupervised Clustering: Since the unsupervised technique did not give good results, we experimented with a non-parametric clustering approach, namely, Cross-Association(Chakrabarti et al., 2004). It is a non-parametric unsupervised clustering algorithm for similarity (boolean) matrices. We obtain the similarity matrix in our domain by thresholding the ROUGE similarity score matrix. This technique also did not give us good clusters, evaluated empirically. The plausible reason for the poor performance seems to be that the technique is based on MDL (Minimum Description Length) principle. Since in our domain we expect a large number of clusters with small membership along many singletons, MDL principle is not likely to perform well.

Deterministic Clustering:

As the unsupervised techniques did not perform well, we tried deterministic clustering based on graph connectivity. The underlying intuition is that all the sentences $X_{1...n}$ that are "similar" to any other sentence Y_i should be in the same cluster even though X_j and X_k may not be "similar" to each other. Thus we find the connected components in the similarity matrix and label them as individual clusters.[4]

[4]This approach is similar to agglomerative single linkage clustering.

We created a similarity matrix by thresholding the ROUGE score. In the event, the clusters obtained by this approach were also not good, evaluated empirically. This led us to revisit the similarity function and tune it. We factored the ROUGE-WLCS score, which is an F-measure score, into its component Precision and Recall scores and experimented with various combinations of using the Precision and Recall scores. We finally chose a combined Precision and Recall measure (not f-measure) in which both the scores were independently thresholded. The motivation for the measure is that in our domain we desire to have high precision matches. Additionally we need to control the length of the sentences in the cluster for which we require a Recall threshold. F-measure (which is the harmonic mean of Precision and Recall) does not give us the required individual control. We set up our experiments such that while comparing two sentences the longer sentence is always treated as the reference and the shorter one as the candidate. This helps us in interpreting the Precision/Recall measures better and thresholding them accordingly. The approach gave us 149 clusters, which looked good on empirical evaluation. We can argue that using this modified similarity function for previous unsupervised approaches could have given better results, but we did not reevaluate those approaches as our aim of getting a reasonable clustering approach is fulfilled with this simple scheme and tuning the unsupervised approaches can be interesting future work.

4.3 Alignment

After obtaining the clusters using the Deterministic approach we needed to find out the template corresponding to each of the cluster. Fairly intuitively we computed the Longest Common Subsequence(LCS) between the sentences in each cluster which we then claim to be the template corresponding to the cluster. This resulted in a set of 149 templates, similar to the Predicate Argument based approach, as shown in figure 6.

5 Results

5.1 Evaluation Scheme

Since there is no standard way to evaluate template extraction for summary creation, we adopted a mix of subjective and automatic measures for evaluating the templates extracted. We define precision for this particular problem as:

$$precision = \frac{number\ of\ domain\ relevant\ templates}{total\ number\ of\ extracted\ templates}$$

This is a subjective measure and we undertook a study involving three subjects who were accustomed to the language used in the corpus. We asked the human subjects to mark each template as relevant or non-relevant to the weather forecast domain. We also asked them to mark the template as grammatical or ungrammatical if it is non-relevant.

Our other metric for evaluation is automatic recall. It is based on using the ROUGE-WLCS metric to determine a match between the preprocessed (baseNP and NE tagged) test corpora with the proposed set of correct templates, a set determined by taking an intersection of only the relevant templates marked by each judge. For the ROUGE based method, the test corpus consists of 262 sentences, while for the predicate-argument based method it consists of a set of 263 propositions extracted from the 262 sentences using ASSERT followed by a filtering of invalid propositions (e.g. ones starting with a verb). Amongst different ROUGE scores (precision/recall/f-measure), we consider precision as the criterion for deciding a match and experimented with different thresholding values.

Main Verb	Precision	Main Verb	Precision
deepen	0.67	weaken	0.83
expect	0.76	lie	0.57
drift	0.93	continue	0.97
build	0.95	fill	0.80
cross	0.78	move	0.86

Table 1: Precision for top 10 most frequently occurring verbs

5.2 Results: Predicate-Argument Based Approach

Table 1 shows the precision values for top 10 most frequently occurring verbs. (Since a major proportion ($> 90\%$) of the templates are covered by these verbs, we don't show all the precision values; it also helps to contain space.) The overall precision value achieved was 84.21%, the inter-rater Fleiss' kappa measure (Fleiss, 1971) between the judges being

$\kappa = 0.69$, demonstrating substantial agreement. The precision values are encouraging, and in most cases the reason for low precision is because of erroneous performance of the semantic role labeler system, which is corroborated by the percentage (47.47%) of ungrammatical templates among the irrelevant ones.

Results for the automated recall values are shown in Figure 8, where precision values are varied to observe the recall. For 0.9 precision in ROUGE-WLCS, the recall is 0.3 which shows that there is a 30% near exact coverage over propositions, while for 0.6 precision in ROUGE-WLCS, the recall is an encouraging 81%.

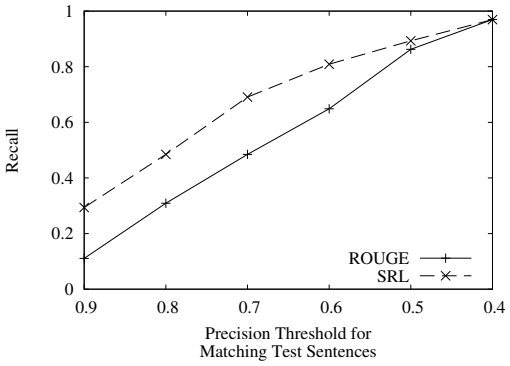

Figure 8: Automated Recall based on ROUGE-WLCS measure comparing the test corpora with the set of templates extracted by the Predicate-Argument (SRL) and the ROUGE based method.

5.3 Results: ROUGE based approach

Various precision and recall thresholds for ROUGE were considered for clustering. We empirically settled on a recall threshold of 0.8 since this produces the set of clusters with optimum number of sentences. The number of clusters and number of sentences in clusters at this recall values are shown in Figure 9 for various precision thresholds.

Precision was measured in the same way as the predicate argument approach and the value obtained was 76.3%, with Fleiss' kappa measure of $\kappa = 0.79$. The percentage of ungrammatical templates among the irrelevant ones was 96.7%, strongly indicating that post processing the templates using a parser can, in future, give substantial improvement. During error analysis, we observed simple grammatical errors in templates; first or last word being prepositions. So a fairly simple error recovery module that strips the leading and trailing prepositions was introduced. 20 templates out of the 149 were modified by the error recovery module and they were evaluated again by the three judges. The precision obtained for the modified templates was 35%, with Fleiss' kappa $\kappa = 1$, boosting the overall precision to 80.98%. The overall high precision is motivating as this is a fairly general approach that does not require any NLP resources. Figure 8 shows the automated recall values for the templates and abstracted sentences from the held-out dataset. For high precision points, the recall is low because there is not an exact match for most cases.

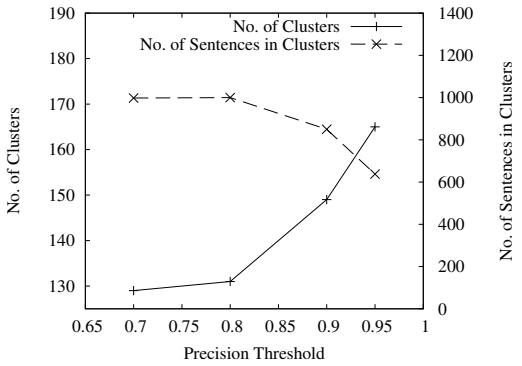

Figure 9: Number of clusters and total number of sentences in clusters for various Precision Thresholds at Recall Threshold=0.8

6 Conclusion and Future Work

In this paper, we described two new approaches for template extraction for briefing generation. For both approaches, high precision values indicate that meaningful templates are being extracted. However, the recall values were moderate and they hint at possible improvements. An interesting direction of future research is merging the two approaches and have one technique benefit from the other. The approaches seem complementary as the ROUGE based technique does not use the structure of the sentence at all whereas the predicate-argument approach is heavily dependent on it. Moreover, the predicate argument based approach gives general templates with one predicate while ROUGE based approach

can extract templates containing multiple verbs. It would also be desirable to establish the generality of the techniques, by using other domains such as newswire, medical reports and others.

Acknowledgements We would like to express our gratitude to William Cohen and Noah Smith for their valuable suggestions and inputs during the course of this work. We also thank the three anonymous reviewers for helpful suggestions. This work was supported by DARPA grant NBCHD030010. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

ACE (2007). Automatic content extraction program. http://www.nist.gov/speech/tests/ace/.

Chakrabarti, D., Papadimitriou, S., Modha, D. S., & Faloutsos, C. (2004). Fully automatic cross-associations. *Proceedings of KDD '04* (pp. 79–88). New York, NY, USA: ACM Press.

Charniak, E. (2001). Immediate-head parsing for language models. *Proceedings of ACL '01* (pp. 116–123).

Collier, R. (1998). *Automatic template creation for information extraction*. Doctoral dissertation, University of Sheffield.

Elhadad, N., Kan, M.-Y., Klavans, J. L., & McKeown, K. (2005). Customization in a unified framework for summarizing medical literature. *Artificial Intelligence in Medicine*, *33*, 179–198.

Fellbaum, C. (1998). *WordNet – An Electronic Lexical Database*. MIT Press.

Filatova, E., Hatzivassiloglou, V., & McKeown, K. (2006). Automatic creation of domain templates. *Proceedings of COLING/ACL 2006* (pp. 207–214).

Fleiss, J. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin* (pp. 378–382).

Kingsbury, P., Palmer, M., & Marcus, M. (2002). Adding semantic annotation to the penn treebank. *Proceedings of the HLT'02*.

Kumar, M., Garera, N., & Rudnicky, A. I. (2007). Learning from the report-writing behavior of individuals. *IJCAI* (pp. 1641–1646).

Lin, C.-Y. (2004). ROUGE: A package for automatic evaluation of summaries. *Proceedings of Workshop on Text Summarization*.

Luhn, H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research Development*, *2*, 159–165.

Marsh, E., & Perzanowski, D. (1998). MUC-7 Evaluation of IE Technology: Overview of Results. *Proceedings of MUC-7*. Fairfax, Virginia.

Onyshkevych, B. (1994). Issues and methodology for template design for information extraction. *Proceedings of HLT '94* (pp. 171–176). Morristown, NJ, USA.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. (2001). Bleu: a method for automatic evaluation of machine translation.

Pradhan, S., Ward, W., Hacioglu, K., Martin, J., & Jurafsky, D. (2004). Shallow semantic parsing using support vector machines. *Proceedings of HLT/NAACL '04*. Boston, MA.

Reiter, E., Sripada, S., Hunter, J., Yu, J., & Davy, I. (2005). Choosing words in computer-generated weather forecasts. *Artif. Intell.*, *167*, 137–169.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. *AAAI/IAAI, Vol. 2* (pp. 1044–1049).

Surdeanu, M., Harabagiu, S., Williams, J., & Aarseth, P. (2003). Using predicate-argument structures for information extraction. *Proceedings of ACL 2003*.

Yangarber, R., Grishman, R., Tapanainen, P., & Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. *Proceedings of the 18th conference on Computational linguistics* (pp. 940–946). Morristown, NJ, USA.