Using Discourse Information for Paraphrase Extraction

Michaela Regneri
Dept. of Computational Linguistics
Saarland University
Saarbrücken, Germany
regneri@coli.uni-saarland.de

Rui Wang
Language Technology Lab
DFKI GmbH
Saarbrücken, Germany
ruiwang@dfki.de

Abstract

Previous work on paraphrase extraction using parallel or comparable corpora has generally not considered the documents' discourse structure as a useful information source. We propose a novel method for collecting paraphrases relying on the sequential event order in the discourse, using multiple sequence alignment with a semantic similarity measure. We show that adding discourse information boosts the performance of sentence-level paraphrase acquisition, which consequently gives a tremendous advantage for extracting phrase-level paraphrase fragments from matched sentences. Our system beats an informed baseline by a margin of 50%.

1 Introduction

It is widely agreed that identifying paraphrases is a core task for natural language processing, including applications like document summarization (Barzilay et al., 1999), Recognizing Textual Entailment (Dagan et al., 2005), natural language generation (Zhao et al., 2010; Ganitkevitch et al., 2011), and machine translation (Marton et al., 2009). As a consequence, many methods have been proposed for generating large paraphrase resources (Lin and Pantel, 2001; Szpektor et al., 2004; Dolan et al., 2004). One of the intuitively appropriate data sources for such collections are parallel or comparable corpora: if two texts are translations of the same foreign document, or if they describe the same underlying scenario, they should contain a reasonable number of sentence pairs that convey the same meaning.

Most approaches that extract paraphrases from parallel texts employ some type of pattern matching: sentences with the same meaning are assumed to share many n-grams (Barzilay and Lee, 2003; Callison-Burch, 2008, among others), many words in their context (Barzilay and McKeown, 2001) or certain slots in a dependency path (Lin and Pantel, 2001; Szpektor et al., 2004). Discourse structure has only marginally been considered for this task: For example, Dolan et al. (2004) extract the first sentences from comparable articles and take them as paraphrases. Another approach (Deléger and Zweigenbaum, 2009) matches similar paragraphs in comparable texts, creating smaller comparable documents for paraphrase extraction.

We believe that discourse structure delivers important information for the extraction of paraphrases. Sentences that play the same role in a certain discourse and have a similar discourse context can be paraphrases, even if a semantic similarity model does not consider them very similar. This extends the widely applied distributional hypothesis to the discourse level: According to the distributional hypothesis, entities are similar if they share similar contexts. In our case, entities are whole sentences, and contexts are discourse units.

Based on this assumption, we propose a novel method for collecting paraphrases from parallel texts using discourse information. We create a new type of parallel corpus by collecting multiple summaries for several TV show episodes. The discourse structures of those summaries are easy to compare: they all contain the events in the same order as they have appeared on the screen. This allows us to take sentence order as event-based discourse structure, which is highly parallel for recaps of the same episode.

In its first step, our system uses a sequence align-

Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 916–927, Jeju Island, Korea, 12–14 July 2012.

ment algorithm combined with a state-of-the-art similarity measure. The approach outperforms informed baselines on the task of sentential paraphrase identification. The usage of discourse information even contributes more to the final performance than the sentence similarity measure.

As second step, we extract phrase-level paraphrase fragments from the matched sentences. This step relies on the alignment algorithm's output, and we show that discourse information makes a big difference for the precision of the extraction. We then add more discourse-based information by preprocessing the text with a coreference resolution system, which results in additional performance improvement.

The paper is structured as follows: first we summarize related work (Sec. 2), and then we give an overview over our perspective on the task and sketch our system pipeline (Sec. 3). The following two sections describe the details of the sentence matching step (Sec. 4) and the subsequent paraphrase fragment extraction (Sec. 5). We present both automatic and manual evaluation of the two system components (Sec. 6). Finally, we conclude the paper and give some hints for future work (Sec. 7).

2 Related Work

Previous paraphrase extraction approaches can be roughly characterized under two aspects: 1) data source and 2) granularity of the output.

Both parallel corpora and comparable corpora have been quite well studied. Barzilay and McKeown (2001) use different English translations of the same novels (i.e., monolingual parallel corpora), while others (Quirk et al., 2004) experiment on multiple sources of the same news/events, i.e., monolingual comparable corpora. Commonly used (candidate) comparable corpora are news articles written by different news agencies within a limited time window (Wang and Callison-Burch, 2011). Other studies focus on extracting paraphrases from large bilingual parallel corpora, which the machine translation (MT) community provides in many varieties. Bannard and Callison-Burch (2005) as well as Zhao et al. (2008) take one language as the pivot and match two possible translations in the other languages as paraphrases if they share a common pivot phrase. As parallel corpora have many alternative ways of expressing the same foreign language concept, large quantities of paraphrase pairs can be extracted.

The paraphrasing task is also strongly related to cross-document event coreference resolution, which is tackled by similar techniques used by the available paraphrasing systems (Bagga and Baldwin, 1999; Tomadaki and Salway, 2005).

Most work in paraphrase acquisition has dealt with sentence-level paraphrases, e.g., (Barzilay and McKeown, 2001; Barzilay and Lee, 2003; Dolan et al., 2004; Quirk et al., 2004). Our approach for sentential paraphrase extraction is related to the one introduced by Barzilay and Lee (2003), who also employ multiple sequence alignment (MSA). However, they use MSA at the sentence level rather than at the discourse level.

We take some core ideas from our previous work on mining script information (Regneri et al., 2010). In this earlier work, we focused on event structures and their possible realizations in natural language. The corpus used in those experiments were short crowd-sourced descriptions of everyday tasks written in bullet point style. We aligned them with a hand-crafted similarity measure that was specifically designed for this text type. In this current work, we target the general task of extracting paraphrases for events rather than the much more specific script-related task. The current approach uses a domain-independent similarity measure instead of a specific hand-crafted similarity score and is thus applicable to standard texts.

From an applicational point of view, sentential paraphrases are difficult to use in other NLP tasks. At the phrasal level, interchangeable patterns (Shinyama et al., 2002; Shinyama and Sekine, 2003) or inference rules (Lin and Pantel, 2001) are extracted. In both cases, each pattern or rule contains one or several slots, which are restricted to certain type of words, e.g., named entities (NE) or content words. They are quite successful in NE-centered tasks, like information extraction, but their level of generalization or coverage is insufficient for applications like Recognizing Textual Entailment (Dinu and Wang, 2009).

The research on *general* paraphrase fragment extraction at the sub-sentential level is mainly based

on phrase pair extraction techniques from the MT literature. Munteanu and Marcu (2006) extract subsentential translation pairs from comparable corpora using the log-likelihood-ratio of word translation probability. Quirk et al. (2007) extract fragments using a generative model of noisy translations. Our own work (Wang and Callison-Burch, 2011) extends the first idea to paraphrase fragment extraction on monolingual parallel and comparable corpora. Our current approach also uses word-word alignment, however, we use syntactic dependency trees to compute *grammatical* fragments. Our use of dependency trees is inspired by the constituent-tree-based experiments of Callison-Burch (2008).

3 Paraphrases and Discourse

Previous approaches have shown that comparable texts provide a good basis for paraphrase extraction. We want to show that discourse structure is highly useful for precise and high-yield paraphrase collection from such corpora. Consider the following (made-up) example:

(1) [House keeps focusing on his aching leg.$_{1.1}$.] [The psychiatrist suggests him to get a hobby $_{1.2}$.] [House joins a cooking class.$_{1.3}$]

(2) [He tells him that the Ibuprofen is not helping with the pain.$_{2.1}$.] [Nolan tells House to take up a hobby.$_{2.2}$] [Together with Wilson he goes to a cookery course.$_{2.3}$]

Read as a whole, it is clear that the two texts describe the same three events, in the same order, and thus, e.g., 1.2 and 2.2 are paraphrases. However, they share very few n-grams, nor named entities. We determine three factors that can help to identify such paraphrases:

1. Consider the **sequence of events**. A system which recognizes that the three sentence pairs occur in the same sequential event order would have a chance of actually matching the sentences.

2. Do **coreference resolution**. To determine which *sentence parts* actually carry the same meaning, pronoun resolution is essential (e.g., to match "suggest him" and "tells House").

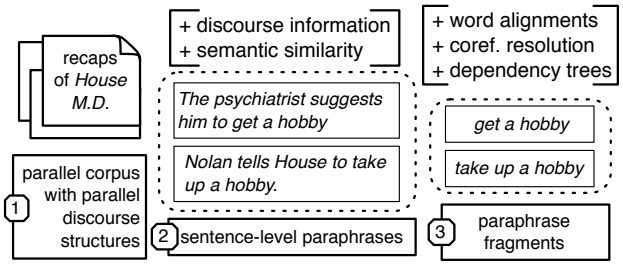

Figure 1: System pipeline

3. Try a generic **sentence similarity model**. Pattern matching or n-gram overlap might not be sufficient to solve this problem.

Our system pipeline is sketched in Fig. 1:

1. **Create a corpus:** First, we create a comparable corpus of texts with *highly comparable discourse structures*. Complete discourse structures like in the RST Discourse Treebank (Carlson et al., 2002) may be very useful for paraphrase computation, however, they are hard to obtain. Discourse annotation is difficult and work-intensive, and full-blown automatic discourse parsers are neither robust nor very precise. To circumvent this problem, we assemble documents that have parallel discourse structures by default: We compile multiple plot summaries of TV show episodes. The textual order of those summaries typically mirrors the underlying event order of the episodes, in the same sequence they happened on screen. We take sentence sequences of recaps as parallel discourse structures.

2. **Extract sentence-level paraphrases:** Our system finds sentence pairs that are either paraphrases themselves, or at least contain paraphrase fragments. This procedure crucially relies on discourse knowledge: A Multiple Sequence Alignment (MSA) algorithm matches sentences if both their inherent *semantic similarities* and the overall similarity score of their *discourse contexts* are high enough.

3. **Extract paraphrase fragments:** Sentence-level paraphrases may be too specific for further domain-independent applications, as they

row	recap 1	recap 2	recap 3	recap 4	recap 5
34	She gives Foreman one shot.	Cuddy tells Foreman he has one chance to prove to her he can run the team.	⊘	Cuddy agrees to give him one chance to prove himself.	Foreman insists he deserves a chance and Cuddy gives in, warning him he gets one shot.
35	⊘	⊘	⊘	Foreman, Hadley, and Taub get the conference room ready and Foreman explains that he'll be in charge.	Foreman gives the news to Thirteen and Taub and they unpack the conference room and go with a diagnosis of CRPS.
36	They decide that it might be CRPS and Foreman orders a spinal stimulation.	⊘	Foreman says to treat him for complex regional pain syndrome with a spinal stimulation.	⊘	⊘

Figure 2: Excerpt from an alignment table for 5 exemplary recaps of Episode 2 (Season 6).

contain specific NEs (e.g. "House") or time references. Thus we take a necessary second step and extract finer-grained paraphrase *fragments* from the sentence pairs matched in step 2. The resulting matched phrases should be grammatical and interchangeable regardless of context. We propose and compare different fragment extraction algorithms.

The remainder of the paper shows how both of the paraphrasing steps benefit from using a corpus with highly parallel discourse structures: The system components employ discourse information either directly by using MSA (step 1) or coreference resolution (step 2), or indirectly, because using MSA in step 1 results in a high precision gain for the subsequent second step.

4 Sentence Matching with MSA

This section explains how we apply MSA to extract sentence-level paraphrases from a comparable corpus. As our input data, we manually collect recaps for *House M.D.* episodes from different sources on the web[1]. *House* episodes have an intermediate length (∼45 min), which results in recaps of a convenient size (40 to 150 sentences). The result is one comparable document collection per episode. We applied a sentence splitter (Gillick, 2009) to the documents and treat them as sequences of sentences for further processing.

Sequence alignment takes as its input two sequences consisting of elements of some alphabet, and an alphabet-specific score function c_m over pairs of sequence elements. For insertions and deletions, the algorithm additionally takes gap costs (c_{gap}). Multiple Sequence Alignment generalizes pairwise alignment to arbitrarily many sequences. MSA has its main application area in bioinformatics, where it is used to identify equivalent parts of DNA (Durbin et al., 1998). Our alphabet consists of sentences, and a sequence is an ordered sentence list constituting a recap.

A Multiple Sequence Alignment results in a table like Fig. 2. Each column contains the sentences of one recap, possibly intermitted with gaps ("⊘"), and each row contains at least one non-gap. If two sentences end up in the same row, they are *aligned*; we take aligned sentence to be paraphrases. Aligning a sentence with a gap can be thought of as an insertion or deletion. Each alignment has a *score* which is the sum of all scores for substitutions and all costs for insertions and deletions. Informally, the alignment

[1] e.g. `http://house.wikia.com` – for a detailed list of URLs, please check the supplementary material or contact the authors.

score is the sum of all scores for each pair of cells (c_1, c_2), if c_1 and c_2 are in the same row. If either c_1 or c_2 is a gap, the pair's score is c_{gap}. If both cells contain sentences, the score is $c_m(c_1, c_2)$.

Fern and Stevenson (2009) showed that sophisticated similarity measures improve paraphrasing, so we apply a state-of-the-art vector space model (Thater et al., 2011) as our score function. The vector space model provides contextualized similarities of words, i.e. the vector of each word is disambiguated by the context the current instance occurs in. $c_m(c_1, c_2)$ returns the model's similarity score for c_1 and c_2.

We re-implement a standard MSA algorithm (Needleman and Wunsch, 1970) which approximates the best MSA given the input sequences, c_m and c_{gap}. This algorithm recursively aligns two sequences at a time, treating the resulting alignment as a new sequence. This does not necessarily result in the globally optimal alignment, because the order in which sequences are aligned can change the final output. Given this constraint, the algorithm finds the best alignment, which - in our case - is the alignment with the maximal score. Intuitively, we are looking for the alignment where the most similar sentences with the most similar preceding and trailing contexts end up as paraphrases.

5 Paraphrase Fragment Extraction

Taking the output of the sentence alignment as input, we next extract shorter phrase-level paraphrases (*paraphrase fragments*) from the matched sentence pairs. We try different algorithms for this step, all relying on word-word alignments.

5.1 Preprocessing

Before extracting paraphrase fragments, we first preprocess all documents as follows:

Stanford CoreNLP [2] provides a set of natural language analysis tools. We use the part-of-speech (POS) tagger, the named-entity recognizer, the parser (Klein and Manning, 2003), and the coreference resolution system (Lee et al., 2011). In particular, the dependency structures of the parser's output are used for VP-fragment extraction (Sec. 5.3). The output from the coreference resolution system is used to cluster all mentions referring to the same entity and to select one as the *representative* mention. If the representative mention is not a pronoun, we modify the original texts by replacing all pronoun mentions in the cluster with the syntactic head of the representative mention. Note that the coreference resolution system is applied to each recap as a whole.

GIZA++ (Och and Ney, 2003) is a widely used word aligner for MT systems. We amend the input data by copying identical word pairs 10 times and adding them as additional 'sentence' pairs (Byrne et al., 2003), in order to emphasize the higher alignment probability between identical words. We run GIZA++ for bi-directional word alignment and obtain a lexical translation table.

5.2 Fragment Extraction

As mentioned in Sec. 2, we choose to use alignment-based approaches to this task, which allows us to use many existing MT techniques and tools. We mainly follow our previous approach (Wang and Callison-Burch, 2011), which is a modified version of an approach by Munteanu and Marcu (2006) on translation fragment extraction. We briefly review the three-step procedure here and refer the reader to the original paper for more details:

1. Establish word-word alignment between each sentence pair using GIZA++;
2. Smooth the alignment based on lexical occurrence likelihood;
3. Extract fragment pairs using different heuristics, e.g., non-overlapping n-grams, chunk boundaries, or dependency trees.

After obtaining a lexical translation table by running GIZA++, for each word pair, w_1 and w_2, we use both positive and negative lexical associations for the alignment, which are defined as the conditional probabilities $p(w_1|w_2)$ and $p(w_1|\neg w_2)$, respectively. The resulting alignment can be further constrained by a modified longest common substring (LCS) algorithm, which takes sequences of

[2] http://nlp.stanford.edu/software/corenlp.shtml

words instead of letters as input. Smoothing (step 2) is done for each word by taking the average score of it and its four neighbor words. All the word alignments (excluding stop-words) with positive scores are selected as candidate fragment elements.

Provided with the candidate fragment elements, we previously (Wang and Callison-Burch, 2011) used a chunker[3] to finalize the output fragments, in order to follow the linguistic definition of a *(para-) phrase*. We extend this step in the current system by applying a dependency parser to constrain the boundary of the fragments (Sec. 5.3). Finally, we filter out trivial fragment pairs, such as identical or the original sentence pairs.

5.3 VP-fragment Extraction

To obtain more grammatical output fragments, we add another layer of linguistic information to our input sentences. Based on the dependency parses produced during preprocessing, we extract phrases containing verbs and their complements. More precisely, we match two phrases if their respective subtrees t_1 and t_2 satisfy the following conditions:

- The subtrees mirror a complete subset of the GIZA++ word alignment, i.e., all words aligned to a given word in t_1 are contained in t_2, and vice versa. For empty alignments, we require an overlap of at least one lemma (ignoring stop words).
- The root nodes of t_1 and t_2 have the same roles within their trees, e.g., we match clauses with an `xcomp`-label only with other `xcomp`-labelled clauses.
- Both t_1 and t_2 contain at least one verb with at least one complement. To enhance recall, we additionally extract complete prepositional phrases.
- We exclude trivial fragment pairs that are prefixes or suffixes of each other (or identical).

The main advantage of this approach lies in the output's grammaticality, because the subtrees always match complete phrases. This method also functions as a filtering mechanism for mistakenly aligned sentences: If only the two sentence nodes are returned as possible matching partners, the pair is discarded from the results.

[3]We use the same OpenNLP chunker (`http://opennlp.sourceforge.net/`) for consistency.

6 Evaluation

We evaluate both sentential paraphrase matching and paraphrase fragment extraction using manually labelled gold standards (provided in the supplementary material). We collect recaps for all 20 episodes of season 6 of *House M.D.*, taking 8 summaries per episode (the supplementary material contains a list of all URLs). This results in 160 documents containing 14735 sentences. For evaluation, we use all episodes except no. 2, which is held out for parameter optimizations and other development purposes.

6.1 Sentential Paraphrase Evaluation

To evaluate sentence matching, we adapt the baselines from our earlier work (Regneri et al., 2010) and create a new gold standard. We compute precision, recall and accuracy of our main system and suggest baselines that separately show the influence of both the MSA and the semantic scoring function.

Gold-Standard

We aim to create an evaluation set that contains a sufficient amount of genuine paraphrases. Finding such sentence pairs with random sampling and manual annotation is infeasible: There are more than $200,000,000$ possible sentence pairs, and we expect less than 1% of them to be paraphrases. We thus sample pairs that either the system or the baselines recognized as paraphrases and try to create an evaluation set that is not biased towards the actual system or any of the baselines. The evaluation set consists of 2000 sentence pairs: 400 that the system recognized as paraphrases, 400 positively labelled pairs for each of the three baselines (described in the following section) and 400 randomly selected pairs. For the final evaluation, we compute precision, recall, f-score and accuracy for our main system and each baseline on this set.

Two annotators labelled each sentence pair (S_1, S_2) with one of the following labels:

1. **paraphrases**: S_1 and S_2 refer to exactly the same event(s).
2. **containment**: S_1 contains all the event information mentioned in S_2, but refers to at least

one additional event, or vice versa.

3. **related**: S_1 and S_2 overlap in at least one event reference, but both refer to at least one additional event.

4. **unrelated**: S_1 and S_2 do not overlap at all.

This scheme has a double purpose: The main objective is judging whether two sentences contain paraphrases (1-3) or if they are unrelated (4). We use this coarser distinction for system evaluation by collapsing the categories 1-3 in one *paraphrase*$_{coll}$ category. Secondly, the annotation shows how well the sentences fit each other's content (1 vs. 2&3), and how much work needs to be done to extract the sentence parts with the same meaning (2 vs. 3).

The inter-annotator agreement according to Cohen's Kappa (Cohen, 1960) is $\kappa = 0.55$ ("moderate agreement"). The distinction between *unrelated* cases and elements of *paraphrase*$_{coll}$ reaches $\kappa = 0.71$ ("substantial agreement"). For the final gold standard, a third annotator resolved all conflict cases.

Among all gold standard sentence pairs, we find 158 *paraphrases*, 238 *containment* cases, 194 *related* ones and 1402 *unrelated*. We had to discard 8 sentence pairs because one of the items was invalid or empty. The high proportion of 'unrelated' cases results from the 400 random pairs and the low precision of the baselines. Looking at the paraphrases, 27% of the 590 instances in the *paraphrase*$_{coll}$ category are proper paraphrases, and 73% of them contain additional information that does not belong to the paraphrased part.

Experimental Setup

We compute precision, recall and f-score with respect to the gold standard (paraphrases are members of *paraphrase*$_{coll}$), taking f-score as follows:

$$f\text{-}score = \frac{2 * precision * recall}{precision + recall}$$

We also compute accuracy as the overall fraction of correct labels (negative and positive ones).

Our main system uses MSA (denoted by MSA afterwards) with vector-based similarities (VEC) as a scoring function. The gap costs are optimized for f-score, resulting in $c_{gap} = 0$.[4]

To show the contribution of MSA's structural component and compare it to the vector model's contribution, we create a second MSA-based system that uses MSA with BLEU scores (Papineni et al., 2002) as scoring function (MSA+BLEU). BLEU establishes the average 1-to-4-gram overlap of two sentences. The gap costs for this baseline were optimized separately, ending up with $c_{gap} = 1$.

In order to quantify the contribution of the alignment, we create a discourse-unaware baseline by dropping the MSA and using a state-of-the-art clustering algorithm (Noack, 2007) fed with the vector space model scores (CLUSTER+VEC). The algorithm partitions the set of sentences into paraphrase clusters such that the most similar sentences end up in one cluster. This does not require any parameter tuning.

We also show a baseline that uses the clustering algorithm with BLEU scores (CLUSTER+BLEU). The comparison of this baseline with the other clustering-baseline that uses vector similarities helps to underline the sentence similarities' advantage compared to pure word overlap. Note that the CLUSTER+BLEU system resembles popular n-gram overlap measures for paraphrase classification.

We also show the results completely random label assignment, which constitutes a lower bound for the baselines and the system.

Results

Overall, our system extracts 20379 paraphrase pairs. Tab. 1 shows the evaluation results on our gold-standard.

The MSA based system variants outperform the two clustering baselines significantly (all levels refer to $p = 0.01$ and were tested with a resampling test (Edgington, 1986)).

The clustering baselines perform significantly better than a random baseline, especially considering recall. The more elaborated vector-space measure even gives 10% more in precision and accuracy, and overall 14% more in f-score. This is al-

[4] Gap costs directly influence precision and recall: "cheap" gaps lead to a more restrictive system with higher precision, and more expensive gaps give more recall. We chose f-score as our objective.

System	*Prec.*	*Recall*	*F-score*	Acc.
RANDOM	0.30	0.49	0.37	0.51
CLUSTER+BLEU	0.35	0.63	0.45	0.54
CLUSTER+VEC	0.40	0.68	0.51	0.61
MSA+BLEU	0.73	**0.74**	**0.73**	0.84
MSA+VEC	**0.79**	0.66	0.72	**0.85**

Table 1: Results for sentence matching.

ready a remarkable improvement compared to the random baseline, and still a significant one compared to CLUSTER+BLEU.

Adding structural knowledge with MSA improves the clustering's accuracy performance by 24% (CLUSTER+VEC vs. MSA+VEC), precision even goes up by 39%.

Intuitively we expected the MSA-based systems to end up with a higher recall than the clustering baselines, because sentences can be matched even if their similarity is moderate or low, but their discourse context is highly similar. However, this is only the case for the system using BLEU scores, but not for the system based on the vector space model. One possible explanation lies in picking f-score as objective for the optimization of the gap costs for MSA: For the naturally more restrictive word overlap measure, this leads to a more recall-oriented system with a low threshold for aligning sentences, whereas the gap costs for the vector-based system favors a more restrictive alignment with more precise results.

The comparison of the two MSA-based systems highlights the great benefit of using structural knowledge: Both MSA+BLEU and MSA+VEC have comparable f-scores and accuracy. The advantage from using the vector-space model that is still obvious for the clustering baselines is nearly evened out when adding discourse knowledge as a backbone. However, the vector model still results in nominally higher precision and accuracy.

It is hard to do a direct comparison with state-of-the-art paraphrase recognition systems, because most are evaluated on different corpora, e.g., the Microsoft paraphrase corpus (Dolan and Brockett, 2005, MSR). We cannot apply our system to the MSR corpus, because we take complete texts as input, while the MSR corpus solely delivers sentence pairs. While the MSR corpus is larger than our collection, the wording variations in its paraphrase pairs are usually lower than for our examples. Thus the final numbers of previous approaches might be vaguely comparable with our results: Das and Smith (2009) present two systems reaching f-scores of 0.82 and 0.83, with a precision of 0.75 and 0.80. Both precision and f-scores of our msa-based systems lie within the same range. Heilman and Smith (2010) introduce a recall-oriented system, which reaches an f-score of 0.81 by a precision of 0.76. Compared to this system, our approach results in better precision values.

All further computations bases on the system using MSA and the vector space model (MSA+VEC), because it achieves the highest precision and accuracy values.

6.2 Paraphrase Fragment Evaluation

We also manually evaluate precision on paraphrase fragments, and additionally describe the *productivity* of the different setups, providing some intuition about the methods' recall.

Gold-Standard

We randomly collect 150 fragment pairs for each of the five system configurations (explained in the following section). Each fragment pair (f_1, f_2) is annotated with one of the following categories:

1. **paraphrases**: f_1 and f_2 convey the same meaning, i.e., they are well-formed and good matches on the content level.
2. **related**: f_1 and f_2 overlap in their meaning, but one or both phrases have additional unmatched information.
3. **irrelevant**: f_1 and f_2 are unrelated.

This labeling scheme again assesses precision as well as paraphrase granularity. For precision rating, we collapse categories 1&2 into one $paraphrase_{coll}$ category. Each pair is labelled by two annotators, who were shown both the fragments and the whole sentences they originate from. Overall, the raters had an agreement of $\kappa = 0.67$ ("substantial agreement"), which suggests that the task was easier than sentence level annotation. The agreement for the

distinction between the *paraphrase*$_{coll}$ categories and *irrelevant* instances reaches a level of $\kappa = 0.88$ (also "substantial agreement"). All conflicts were again adjudicated by a third annotator. Overall, the gold standard contains 190 paraphrases, 258 *related* pairs and 302 *irrelevant* instances. Unlike previous approaches to fragment extraction, we do not evaluate *grammaticality*, given that the VP-fragment method implicitly constrains the output fragments to be complete phrases.

Configurations & Results

We take the output of the sentence matching system MSA+VEC as input for paraphrase fragment extraction. As detailed in Sec. 5, our core fragment module uses the word-word alignments provided by GIZA++ and uses a chunker for fragment extraction. We successively enrich this core module with more information, either by longest common substring (LCS) matching or by operating on dependency trees (VP). In addition, we evaluate the influence of coreference resolution by preprocessing the input to the best performing configuration with pronoun resolution (COREF).

We mainly compute precision for this task, as the recall of paraphrase fragments is difficult to define. However, we do include a measure we call *productivity* to indicate the algorithm's completeness. It is defined as the ratio between the number of resulting fragment pairs and the number of sentence pairs used as input.

Extraction Method	*Precision*	*Productivity*
MSA	0.57	**0.76**
MSA+LCS	0.45	0.30
MSA+VP	0.81	0.42
MSA+VP+COREF	**0.84**	0.45

Table 2: Results of paraphrase fragment extraction.

Tab. 2 shows the evaluation results. We reach our best precision by using the VP-fragment heuristics, which is still more productive than the LCS method. The grammatical filter gives us a higher precision compared to the purely alignment-based approaches. Enhancing the system with coreference resolution raises the score even further. We cannot directly compare this performance to other systems, as all other approaches have different data sources. However, precision is usually manually evaluated, so the figures are at least indicative for a comparison with previous work: One state-of-the-art system introduced by Zhao et al. (2008) extracts paraphrase fragments from bilingual parallel corpora and reaches a precision of 0.67. We found the same number using our previous approach (Wang and Callison-Burch, 2011), which is roughly equivalent to our core module. Our approach outperforms both by 17% with similar estimated productivity.

As a final comparison, we show how the performance of the sentence matching methods directly affects the fragment extraction. We use the VP-based fragment extraction system (VP), and compare the performances by using either the outputs from our main system (MSA+VP) or alternatively the baseline that replaces MSA with a clustering algorithm (CLUSTER+VP). Both variants use the vector-based semantic similarity measure.

Sentence matching	*Precision*	*Productivity*
CLUSTER+VP	0.31	0.04
MSA+VP	**0.81**	**0.42**

Table 3: Impact of MSA on fragment extraction

As shown in Tab. 3, the precision gain from using MSA becomes tremendous during further processing: We beat the baseline by 50% here, and productivity increases by a factor of 10. This means that the baseline produces on average 0.01 good fragment pairs per matched sentence pair, and the final system extracts 0.3 of them. Those numbers show that for any application that acquires paraphrases of arbitrary granularity, sequential event information provides an invaluable source to achieve a lean paraphrasing method with high precision.

6.3 Example output

Fig. 3 shows exemplary results from our system pipeline, using the VP–FRAGMENTS method with full coreference resolution on the sentence pairs extracted by MSA. The results reflect the importance of discourse information for this task: Sentences are correctly matched in spite of not having common de-

	Sentence 1 [*with fragment 1*]	**Sentence 2 [*with fragment 2*]**
1	Taub meets House for dinner and claims [*that Rachel had a pottery class*].	Taub shows up for his dinner with House without Rachel, explaining [*that she's at a ceramics class*].
2	House doesn't want her to go and she doesn't want to go either, but [*she can't leave her family.*]	Lydia admits that she doesn't want to leave House but [*she has to stay with her family*].
3	Thirteen is in a cab to the airport when she finds out that [*her trip had been canceled*].	Hadley discovers that [*her reservation has been cancelled*].
4	Nash asks House [*for the extra morphine*].	The patient is ready [*for more morphine*].
5	House comes in to tell Wilson that Tucker has cancer and [*shows him the test results*].	House comes in and [*informs Wilson that the tests have proven positive*]: Tucker has cancer.
6	Foreman tells him [*to confide in Cameron*].	When Chase points out they can't move Donny without alerting Cameron, Foreman tells Chase [*to be honest with his wife*].
7	Thirteen breaks [*into the old residence*] and tells Taub that she realizes that he's been with Maya.	Taub and Thirteen break [*into Ted's former residence*].
8	He finds [*a darkened patch on his right foot near the big toe*].	House finally finds [*a tumorous mole on his toe*].

Figure 3: Example results; fragments extracted from aligned sentences are bracketed and *emphasized*.

pendency patterns (e.g., Example 4) or sharing many n-grams (6-8). Additionally, the coreference resolution allows us to match *Rachel* (1) and *Wilson* (5) to the correct corresponding pronouns. All examples show that this technique of matching sentence could even help to make coreference resolution better, because we can easily identify *Cameron* with *his wife*, *Lydia* with the respective pronouns, *Nash* with *The Patient* or the nickname *Thirteen* with *Hadley*, the character's actual name.

7 Conclusion and Future Work

We presented our work on paraphrase extraction using discourse information, on a corpus consisting of recaps of TV show episodes. Our approach first uses MSA to extract sentential paraphrases, which are then further processed to compute finer-grained paraphrase fragments using dependency trees and pronoun resolution. The experimental results show great advantages from using discourse information, beating informed baselines and performing competitively with state-of-the-art systems.

For future work, we plan to use MSA to align single clauses rather than whole sentences. This can also help to define the fragment boundaries more clearly. Additionally, we plan to generalize the method for other parallel texts by preprocessing them with a temporal classifier. In a more advanced step, we will also use the aligned paraphrases to help resolving discourse structure, e.g. for coreference resolution, which could lead to a high-performance bootstrapping system. In a long-term view, it would be interesting to see how aligned discourse trees could help to extract paraphrases from arbitrary parallel text.

Acknowledgements

The first author was funded by the Cluster of Excellence "Multimodal Computing and Interaction" in the German Excellence Initiative. The second Author was funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 287923 (EXCITEMENT, `http://www.excitement-project.eu/`). – We want to thank Stefan Thater for supplying the semantic similarity scores of his algorithm for our data. We are grateful to Manfred Pinkal, Alexis Palmer and three anonymous reviewers for their helpful comments on previous versions of this paper.

References

Amit Bagga and Breck Baldwin. 1999. Cross-document event coreference: annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and its Applications*.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of ACL 2005*.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proc. of HLT-NAACL 2003*.

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proc. of ACL 2001*.

Regina Barzilay, Kathleen McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of ACL 1999*.

W. Byrne, S. Khudanpur, W. Kim, S. Kumar, P. Pecina, P. Virga, P. Xu, and D. Yarowsky. 2003. The Johns Hopkins University 2003 Chinese-English machine translation system. In *Proceedings of the MT Summit IX*.

Chris Callison-Burch. 2008. Syntactic constraints on paraphrases extracted from parallel corpora. In *Proceedings of EMNLP 2008*.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2002. RST Discourse Treebank. LDC.

J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *MLCW*, pages 177–190.

D. Das and N. A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of ACL-IJCNLP 2009*.

Louise Deléger and Pierre Zweigenbaum. 2009. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proceedings of the ACL-IJCNLP BUCC-2009 Workshop*.

Georgiana Dinu and Rui Wang. 2009. Inference rules and their application to recognizing textual entailment. In *Proceedings of EACL 2009*.

W. B. Dolan and C. Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the third International Workshop on Paraphrasing*.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of COLING 2004*.

Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological Sequence Analysis*. Cambridge University Press.

Eugene S Edgington. 1986. *Randomization tests*. Marcel Dekker, Inc., New York, NY, USA.

Samuel Fern and Mark Stevenson. 2009. A semantic similarity approach to paraphrase detection. In *Proceedings of the Computational Linguistics UK (CLUK 2008) 11th Annual Research Colloquium*.

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

Dan Gillick. 2009. Sentence boundary detection and the problem with the u.s. In *Proceedings of HLT-NAACL 2009: Companion Volume: Short Papers.*

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Proceedings of NAACL-HLT 2010*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL 2003*.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the CoNLL-2011 Shared Task*.

Dekang Lin and Patrick Pantel. 2001. DIRT - Discovery of Inference Rules from Text. In *Proceedings of the ACM SIGKDD*.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved Statistical Machine Translation Using Monolingually-Derived Paraphrases. In *Proceedings of EMNLP 2009*.

Dragos Stefan Munteanu and Daniel Marcu. 2006. Extracting Parallel Sub-Sentential Fragments from Non-Parallel Corpora. In *Proceedings of ACL 2006*.

Saul B. Needleman and Christian D. Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3), March.

Andreas Noack. 2007. Energy models for graph clustering. *Journal of Graph Algorithms and Applications*, 11(2):453–480.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*.

Chris Quirk, Chris Brockett, and William B. Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of EMNLP 2004*.

Chris Quirk, Raghavendra Udupa, and Arul Menezes. 2007. Generative models of noisy translations with applications to parallel fragment extraction. In *Proceedings of MT Summit XI*, Copenhagen, Denmark.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. 2010. Learning Script Knowledge with Web Experiments. In *Proceedings of ACL 2010*.

Yusuke Shinyama and Satoshi Sekine. 2003. Paraphrase acquisition for information extraction. In *Proceedings of the ACL PARAPHRASE '03 Workshop*.

Yusuke Shinyama, Satoshi Sekine, and Kiyoshi Sudo. 2002. Automatic paraphrase acquisition from news articles. In *Proceedings of HLT 2002*.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling Web-based Acquisition of Entailment Relations. In *Proceedings of EMNLP 2004*.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. 2011. Word Meaning in Context: A Simple and Effective Vector Model. In *Proceedings of IJCNLP 2011*.

Eleftheria Tomadaki and Andrew Salway. 2005. Matching verb attributes for cross-document event coreference. In *Proc. of the Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*.

Rui Wang and Chris Callison-Burch. 2011. Paraphrase fragment extraction from monolingual comparable corpora. In *Proc. of the ACL BUCC-2011 Workshop*.

Shiqi Zhao, Haifeng Wang, Ting Liu, and Sheng Li. 2008. Pivot Approach for Extracting Paraphrase Patterns from Bilingual Corpora. In *Proceedings of ACL 2008*.

Shiqi Zhao, Haifeng Wang, Xiang Lan, and Ting Liu. 2010. Leveraging Multiple MT Engines for Paraphrase Generation. In *Proceedings of COLING 2010*.