Language Resources and Evaluation manuscript No.
(will be inserted by the editor)

Parser Evaluation Using Textual Entailments

Deniz Yuret . Laura Rimell . Aydın Han

Received: date / Accepted: date

Abstract Parser Evaluation using Textual Entailments (PETE) is a shared task in the SemEval-2010 Evaluation Exercises on Semantic Evaluation. The task involves recognizing textual entailments based on syntactic information alone. PETE introduces a new parser evaluation scheme that is formalism independent, less prone to annotation error, and focused on semantically relevant distinctions. This paper describes the PETE task, gives an error analysis of the top-performing Cambridge system, and introduces a standard entailment module that can be used with any parser that outputs Stanford typed dependencies.

Keywords Parsing · Textual Entailments

1 Introduction

Parser Evaluation using Textual Entailments (PETE) is a shared task in the SemEval-2010 Evaluation Exercises on Semantic Evaluation that involves recognizing textual entailments to evaluate parser performance. Given two text

Deniz Yuret and Aydın Han
Koc University
Rumelifeneri Yolu
34450 Sarıyer, İstanbul, Turkey
Tel.: +90-212-338-1724
Fax: +90-212-338-1548
E-mail: dyuret,ahan@ku.edu.tr

Laura Rimell
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD, UK
Tel.: +44 (0)1223 334696
E-mail: laura.rimell@cl.cam.ac.uk

fragments called “text” (T) and “hypothesis” (H), recognizing textual entailment (RTE) is the task of determining whether the meaning of the hypothesis is entailed (can be inferred) from the text. In contrast to general RTE tasks (Dagan et al 2009) PETE is a targeted textual entailment task that focuses on syntactic entailments:

Text: The man with the hat was tired.
Hypothesis-1: The man was tired. (yes)
Hypothesis-2: The hat was tired. (no)

By syntactic entailments we mean entailments that can be recognized using grammatical knowledge alone, without recourse to background knowledge or logical reasoning. The main goal of PETE is not to create a general entailment system, but to use entailments as a probe to evaluate a basic linguistic competence, in this case identification of grammatical relations.

The participants were provided with a number of text – hypothesis sentence pairs as input (similar to the Text–Hypothesis-1 pair given above). The goal of the participating systems was to output an accurate YES/NO decision on the syntactic entailment status of each pair (eg YES for the Text–Hypothesis-1 pair and NO for the Text–Hypothesis-2 pair). Each entailment was focused on the relationship of a content word pair (eg man–tired for Hypothesis-1 and hat–tired for Hypothesis-2), however these content word pairs were not made available during testing. Table 1 provides some text–hypothesis examples from the actual test set. Section 2 provides further details on the dataset and Section 3 describes the participating systems and their results. All task relevant data is available at http://pete.yuret.com.

Text	Hypothesis	Ent.
There's a man with a wooden leg named Smith	The man is named Smith.	YES
There's a man with a wooden leg named Smith	The leg is named Smith.	NO
Two share a house almost devoid of furniture	A house is shared.	YES
They wanted to touch the mystery.	They wanted the mystery.	NO
It took me five hours to write it that way.	Something took hours.	YES
Add things as you find you need'em.	The things find something.	NO

Table 1 Some text–hypothesis pair examples from the PETE test set. The third column gives the entailment status and the relevant content words are marked in bold.

1.1 Motivation

The motivation behind using targeted textual entailments as a test of linguistic competence is to use non-expert, native speaker judgements and achieve high inter-annotator agreement. It is generally difficult to achieve high inter-annotator agreement in artificial tagging tasks. We cite two examples: Dick-

insen and Meurers (2003) have found that of the 34,564 constituent strings that appear multiple times in Penn Treebank (Marcus et al 1994), 5,384 (16%)

Page 3

have multiple conflicting annotations, of which an estimated 3,934 are errors. If indicative of the general level of inconsistency, 16% is a very high number given that state of the art parsers claim F-scores above 90% (Charniak and Johnson 2005). In the field of word sense disambiguation, Snyder and Palmer (2004), the organizers of the "English all-words task" in Senseval-3, state: "it seems that the best systems have hit a wall in the 65-70% range. This is not surprising given the typical inter-annotator agreement of 70-75% for this task."

The reason for the low inter-annotator agreement is usually not the annotators' lack of comprehension of the example sentences. The problem is their less than perfect ability to be consistent with the annotation guidelines, or the difficulty of coming up with consistent guidelines in the first place: The Penn Treebank annotation guidelines exceed 400 pages. WordNet (used in Senseval-3) defines more than 100,000 word senses, some of which are difficult to distinguish even for professional lexicographers. We believe the situation might be improved if our systems can target the natural competence of the annotators in comprehending sentences rather than their imperfect performance on artificial annotation tasks.

One can envision annotation tasks that probe the ability to identify grammatical relations, word senses, co-references etc. using basic sentence comprehension and generation skills rather than relying on detailed annotation guidelines or sense inventories. One example that targets the natural competence of annotators in comprehending and generating sentences is the lexical substitution task (McCarthy and Navigli 2007) introduced in SemEval-2007. Unlike standard word sense disambiguation tasks where the annotators need to comprehend not only the example sentences but the dictionary definitions of the target words, the lexical substitution task asks them to come up with substitutes for the target word that preserve the meaning of the sentence. Another example is (Erk et al 2009), which introduces a usage similarity task asking the annotators to judge the similarity between different usages of a word without relying on a sense inventory. Targeted textual entailment tasks that focus on one type of linguistic competence, like PETE, may be one possible path in the direction of more such evaluation schemes. To our knowledge, PETE is the first task to use crowd-sourcing with non-expert, native speaker judgements for parser evaluation, which has traditionally had to rely on trained experts because of the complexity of labeling parse trees.

1.2 Other approaches

There are currently two main approaches in the field of parser evaluation.

The treebank based measures introduced nearly two decades ago (Black et al 1991) compare phrase-structure bracketings or dependency links produced by the parser with the ones in an annotated corpus, or "treebank". A second, more recent strand of parser evaluation methods is based on grammatical dependency relations, proposed for ease of use by end users and suitable for parser evaluation. These include the grammatical relations (GR) of Carroll

Page 4

et al (1999), the PARC representation (King et al 2003), and Stanford typed dependencies (SD) (De Marneffe et al 2006) (see Bos et al (2008) for other proposals).

Compared to the first approach, treebank based evaluation methods, parser evaluation using short textual entailments has the following advantages:

Consistency: Recognizing syntactic entailments is a more natural task for people than treebank annotation. Focusing on a natural human competence makes it practical to collect high quality evaluation data from untrained annotators. The PETE dataset was annotated by untrained Amazon Mechanical Turk workers at an insignificant cost ($19.50) and each annotation is based on the unanimous agreement of at least three workers.

Relevance: PETE automatically focuses attention on semantically relevant phenomena rather than differences in annotation style or linguistic convention. Whether a phrase is tagged adjp vs advp rarely affects semantic interpretation. Attaching the wrong subject to a verb or the wrong prepositional phrase to a noun, however, changes the meaning of the sentence. Standard treebank based evaluation metrics do not distinguish between semantically relevant and irrelevant errors (Bonnema et al 1997). In PETE semantically relevant differences lead to different entailments, semantically irrelevant differences do not.

Framework independence: Entailment recognition is a formalism independent task. A common evaluation method for parsers that do not use the Penn Treebank formalism is to automatically convert the Penn Treebank to the appropriate formalism and to perform treebank based evaluation (Nivre et al 2007a; Hockenmaier and Steedman 2007). However, such conversions are noisy due to fundamental cross-formalism differences as well as inconsistencies in the original treebank (Hockenmaier 2003; Cer et al 2010), compounding the already mentioned problems of treebank based evaluation. Clark and Curran (2007) similarly found an upper bound of 85% accuracy when translating between two grammatical dependency based formalisms for parser evaluation. In addition, manually designed treebanks do not naturally lend themselves to unsupervised parser evaluation. Unlike treebank based evaluation, PETE can compare phrase structure parsers, dependency parsers, unsupervised parsers and other approaches on an equal footing.

The second approach, evaluation methods based on grammatical dependency relations, uses a set of binary relations between words in a sentence as the primary unit of representation. These methods share some common motivations: usability by people who are not (computational) linguists and suitability for relation extraction applications. Furthermore, they more closely represent semantics than treebank constituency or dependency measures, and various types of parsers are capable of producing dependencies as an interchange format. Here is an example sentence and its SD representation (De Marneffe and Manning 2008):

Page 5

> Bell, based in Los Angeles, makes and distributes electronic, computer and building products.
>
> nsubj(makes-8, Bell-1)
> nsubj(distributes-10, Bell-1)
> partmod(Bell-1, based-3)
> nn(Angeles-6, Los-5)
> prep-in(based-3, Angeles-6)
> conj-and(makes-8, distributes-10)
> amod(products-16, electronic-11)
> conj-and(electronic-11, computer-13)
> amod(products-16, computer-13)
> conj-and(electronic-11, building-15)
> amod(products-16, building-15)
> dobj(makes-8, products-16)

PETE was inspired by such methods, but goes one step further by translating most of these dependencies into natural language entailments:

> Bell makes something.
> Bell distributes something.
> Someone is based in Los Angeles.
> Someone makes products.

PETE has some advantages over representations based on grammatical relations:

Ease of use: Each of the three proposals of grammatical dependency relations mentioned in this paper (GR, PARC, SD) uses a different set of binary relations. For example SD defines 55 relations organized in a hierarchy, and it may be non-trivial for a non-linguist to understand the difference between ccomp (clausal complement with internal subject) and xcomp (clausal complement with external subject) or between nsubj (nominal subject) and

nsubj (controlling subject). In this study we were able to achieve unani- mous agreement among 3 or more untrained annotators for close to half of the entailments generated for PETE without any training, correction or adjudication.

Relevance: Though grammatical dependency schemes represent more semantic information than treebank annotation, they still give equal weight to dependencies of differing semantic importance, for example determiners compared to verbal arguments. They are typically used in the aggregate, so that evaluation is weighted towards more frequent dependency types, not necessarily more important ones.[1] When labeled dependencies are used, differences in annotation style can affect evaluation, for example whether

1 The collapsed and propagated version of Stanford dependencies somewhat mitigates this problem and this is the parser output representation we chose to use as input to the example entailment module of Section 7.

a dependency is labeled amod or advmod. In contrast, PETE is designed to focus on semantically relevant types, and is label-free.

1.3 Challenges

There are also significant challenges associated with an evaluation scheme like PETE. It is not always clear how to convert certain relations into grammatical hypothesis sentences without including most of the original sentence in the hypothesis. Including too much of the sentence in the hypothesis would increase the chances of getting the right answer with the wrong parse. Grammatical hypothesis sentences are especially difficult to construct when a (negative) entailment is based on a bad parse of the sentence. Introducing dummy words like “someone” or “something” alleviates part of the problem but does not help in the case of clausal complements. In summary, PETE makes the annotation phase more practical and consistent but shifts the difficulty to the entailment creation phase.

PETE gets closer to an extrinsic evaluation by focusing on semantically relevant, application oriented differences that can be expressed in natural language sentences. This makes the evaluation procedure indirect: a parser developer has to write an extension that can handle entailment questions. However, given the simplicity of the entailments, the complexity of such an extension is comparable to one that extracts grammatical relations. In Section 7 we present a standard entailment module which can be used with any parser that can output Stanford typed dependencies.

The balance of what is being evaluated is also important. A treebank based evaluation scheme may mix semantically relevant and irrelevant mistakes, but

at least it covers every sentence at a uniform level of detail. In this evaluation, we focused on sentences and relations where state of the art parsers make mistakes. We hope this methodology will uncover weaknesses that the next generation of parsers can focus on.

1.4 Summary

The remaining sections will go into more detail about these challenges and the solutions we have chosen to implement. Section 2 explains the method followed to create the PETE dataset. Section 3 presents the participating systems, their methods and results. Section 4 presents the best scoring Cambridge system in more detail. Sections 5 and 6 give a detailed error analysis of the c&c parser and the entailment system used in the Cambridge system. Section 7 introduces a standard entailment system for Stanford typed dependencies and evaluates some example systems for several state of the art parsers. Section 8 summarizes our contribution.

Page 7

2 Dataset

To generate the entailments for the PETE task we used the following three steps:

– Identify syntactic dependencies challenging to state of the art parsers.
– Construct short entailment sentences that paraphrase those dependencies.
– Identify the subset of the entailments with high inter-annotator agreement.

2.1 Identifying Challenging Dependencies

To identify syntactic dependencies that are challenging for current state of the art parsers, we used example sentences from the following sources:

– The “Unbounded Dependency Corpus” (Rimell et al 2009). An unbounded dependency construction contains a word or phrase which appears to have been moved, while being interpreted in the position of the resulting “gap”. For example, the relation between wrote and paper in the paper we wrote is an example of extraction from a reduced relative clause. An unlimited number of clause boundaries may intervene between the moved element and the gap (hence “unbounded”).
– A list of sentences from the Penn Treebank on which the Charniak parser (Charniak and Johnson 2005) performs poorly 2.
– The Brown section of the Penn Treebank.

We tested a number of parsers (both phrase structure and dependency) on these sentences and identified the differences in their output. We took sentences where at least one of the parsers gave a different answer than the gold parse. (The gold parses were available since all sentences came from existing treebanks.) Some of these differences reflected linguistic convention rather than semantic disagreement (eg representation of coordination) and some did not represent meaningful differences that can be expressed with entailments (eg labeling a phrase adjp vs advp). The remaining differences typically reflected genuine semantic disagreements that would affect downstream applications. These were chosen to turn into entailments in the next step.

2.2 Constructing Entailments

Entailment construction was performed manually by annotators trained to interpret phrase structure and dependency parser outputs. Each hypothesis sentence was based on the relationship between two content words that have a syntactic dependency. The content word pairs were chosen to demonstrate differences between a parser output and the gold parse. All true and false hypotheses generated in this fashion that passed the annotator agreement test

2 http://www.cs.brown.edu/ec/papers/badPars.txt.gz

as described in Section 2.3, were added to the dataset. The instructions for the annotators were:

1. Idenfity a sentence where at least one parser gives a different parse tree than the gold parse. Use this sentence as the text of a text–hypothesis pair.
2. Identify content words (defined as nouns, verbs, adjectives, and adverbs) in the sentence which have different syntactic heads or different relations to their syntactic heads in the parser output and the gold parse. If the syntactic head is a function word then consider the closest content word ancestor.
3. For each word–head pair identified in the previous step, construct a minimal hypothesis sentence that expresses the same syntactic relation between the two as was observed in the source parse tree. If the pair comes from the gold parse this generates a TRUE entailment, otherwise this generates a FALSE entailment.
4. If the two content words are not sufficient to construct a grammatical sentence use one of the following techniques:
   – Complete the mandatory elements using the words “somebody” or “something” (eg to express the subject-verb dependency in “John kissed Mary.” construct the hypothesis “John kissed somebody.”).
   – Make a passive sentence to avoid using a spurious subject (eg to ex-

press the verb-object dependency in “John kissed Mary.” construct the hypothesis “Mary was kissed.”).

– Make a copular sentence or use existential “there” to express noun modification (eg to express the noun-modifier dependency in “The big red boat sank.” construct the hypothesis “The boat was big.” or “There was a big boat.”).

As an example consider the sentence “John slept in the bed.” Let us consider what entailments can be constructed from this sentence, assuming we have the gold parse tree. The three content words are “John”, “slept”, and “bed”. “Slept” is the root of the sentence and has no head. The head of “John” is “slept”, so we generate the hypothesis (John slept). The syntactic head of “bed” is “in”, a function word, so we include the content word ancestor “slept”, resulting in the hypothesis (Somebody slept in the bed). Note that we introduce “Somebody”, as in step 4 of the instructions, to make the hypothesis a complete sentence without including more constituents from the text. These are the only two hypothesis sentences that can be generated for this sentence.

In general the number of content words gives an upper bound on the number of entailments (text–hypothesis pairs) generated from a sentence. However not every entailment generated in this way made it into the final dataset because we only included entailments related to parser errors, as described in the previous section, and we filtered ones that did not result in unanimous annotator agreement as described in the next section.

The emphasis on having two content words per entailment reflects the relationship between PETE and grammatical dependency schemes. Entailments were constructed manually, although in the future we hope to develop

automatic methods. In the first edition of PETE we did not measure inter-annotator agreement on entailment generation; based on the guidelines there should in general be a single well-defined H for each pair of content words, although issues such as embedded clauses and noun modification may need to be more carefully analyzed from an entailment generation perspective in the future to be sure of this.

A list of the content word pairs used to construct the entailments was provided to participants as background information, but the list was not accessible to the entailment systems developed by the participants. Thus entailment decisions could not be facilitated by limiting interrogation of parser output to specific lexical items (although some systems did independently choose to prioritize general categories of words or relations in their decisions).

2.3 Filtering Entailments

To identify the entailments that are clear to human judgement we used the

following procedure:

– Each entailment was tagged by 5 untrained annotators from the Amazon Mechanical Turk crowd-sourcing service.
– The results from the annotators whose agreement with the "silver" standard truth values fell below 70% were eliminated.
– The entailments for which there was unanimous agreement of at least 3 annotators were kept.

The second step was necessary to eliminate annotators that were answering questions randomly. The "silver" standard truth values were "yes" for entailments generated from parses agreeing with the gold parses, and "no" for entailments generated from incorrect parses (recall that the gold parses were available from existing treebanks). We call these truth values silver rather than gold since they became part of the gold standard only when unanimously agreed to by three annotators. Though not perfect, the 70% measure provided a simple benchmark to detect annotators answering randomly.

The annotators were allowed to give "Not sure" answers which were later grouped with the "No" answers during evaluation. The instructions for the annotators were brief and targeted people with no linguistic background:

> Computers try to understand long sentences by dividing them into a set of short facts. You will help judge whether the computer extracted the right facts from a given set of 25 English sentences. Each of the following examples consists of a sentence (T), and a short statement (H) derived from this sentence by a computer. Please read both of them carefully and choose "Yes" if the meaning of (H) can be inferred from the meaning of (T). Here is an example:
>
> (T) Any lingering suspicion that this was a trick Al Budd had thought up was dispelled.
> (H) The suspicion was dispelled. Answer: YES
> (H) The suspicion was a trick. Answer: NO
>
> You can choose the third option "Not sure" when the (H) statement is unrelated, unclear, ungrammatical or confusing in any other manner.

2.4 Dataset statistics

The final dataset contained 367 entailments which were randomly divided into a 66 sentence development set and a 301 sentence test set. 52% of the entailments in the test set were positive.

Approximately half of the final entailments were based on sentences from the Unbounded Dependency Corpus, a third were from the Brown section of the Penn Treebank, and the remainder were from the Charniak sentences. Table 2 gives the breakdown of the original list of entailments and the ones retained after the annotation filtering, according to the text source and the entailment value. Between 1/3 and 1/2 of the original entailments were kept in the final dataset in each category. 3

	Pre-filter			Post-filter		
	All	Y	N	All	Y	N
Unbounded	529	327	202	196	108	88
Brown	335	211	124	124	61	63
Charniak	116	65	51	47	22	25
Total	980	603	377	367	191	176

Table 2 Breakdown of data by source and entailment value before and after the annotation filter.

Table 3 lists the most frequent grammatical relations and constructions encountered in the entailments before and after the annotation filter. Note that the resolution of each entailment may rely on multiple grammatical phenomena, thus the numbers add up to more than 100%.

GR	Pre-filter	Post-filter
Direct object	48%	42%
Nominal subject	44%	33%
Reduced relative clause	25%	21%
Relative clause	20%	14%
Passive nominal subject	17%	7%
Open clausal complement	6%	2%
Clausal complement	6%	2%
Prepositional modifier	6%	5%
Adverbial modifier	2%	3%
Object of preposition	2%	5%

Table 3 Most frequent grammatical relations and constructions encountered in the entailments before and after the annotation filtering process.

The two groups of entailments that most often failed the inter-annotator agreement filter involved clausal complements and passivization. Constructing

3 Note that some of the difficult constructions, plus noise in the laypeople's responses meant a large percentage of potential entailments didn't pass the filter, but nevertheless at a nominal cost we were able to create a dataset where all the entailments were unanimously agreed by 3 people, which is not the case for most other commonly used treebanks.

entailments for clausal complements based on two content words as described in Section 2.2 was sometimes challenging for the entailment generators and confusing to the annotators. Annotators failed to reach a unanimous agreement on examples like the following:

Text: He found a jar of preserved tomatoes and one of eggs that they had meant to save.
Hypothesis: Somebody had meant to save one.

Passivization, which was used when constructing entailments from a verb object pair also proved to be confusing at times. Annotators also failed to reach unanimous agreement on some examples like the following:

Text: But he kept Fruit of the Loom Inc., the underwear maker that he still controls and serves as chairman and chief executive.
Hypothesis: The maker is served.

3 Task Results

System	Accuracy	Precision	Recall	F1
360-418-Cambridge	0.7243	0.7967	0.6282	0.7025
459-505-SCHWA	0.7043	0.6831	0.8013	0.7375
473-568-MARS-3	0.6678	0.6591	0.7436	0.6988
372-404-MDParser	0.6545	0.7407	0.5128	0.6061
372-509-MaltParser	0.6512	0.7429	0.5000	0.5977
473-582-MARS-5	0.6346	0.6278	0.7244	0.6726
166-415-JU-CSE-TASK12-2	0.5781	0.5714	0.7436	0.6462
166-370-JU-CSE-TASK12	0.5482	0.5820	0.4551	0.5108
390-433-Berkeley Parser Based	0.5415	0.5425	0.7372	0.6250
473-566-MARS-1	0.5282	0.5547	0.4551	0.5108
473-569-MARS-4	0.5249	0.5419	0.5385	0.5402
390-431-Brown Parser Based	0.5216	0.5349	0.5897	0.5610
473-567-MARS-2	0.5116	0.5328	0.4679	0.4983
363-450-VENSES	0.5083	0.5220	0.6090	0.5621
473-583-MARS-6	0.5050	0.5207	0.5641	0.5415
390-432-Brown Reranker Parser Based	0.5017	0.5217	0.4615	0.4898
390-435-Berkeley with substates	0.5017	0.5395	0.2628	0.3534
390-434-Berkeley with Self Training	0.4983	0.5248	0.3397	0.4125
390-437-Combined	0.4850	0.5050	0.3269	0.3969
390-436-Berkeley with Viterbi Decoding	0.4784	0.4964	0.4359	0.4642

Table 4 Participating systems and their scores. The system identifier consists of the participant ID, system ID, and the system name given by the participant. Accuracy gives the percentage of correct entailments. Precision, Recall and F1 are calculated for positive entailments.

Twenty systems from 7 teams participated in the PETE task. Table 4 gives the percentage of correct answers for each system. Twelve systems performed above the “always yes” baseline of 51.83%.

Most systems started the entailment decision process by extracting syntactic dependencies, grammatical relations, or predicates by parsing the text and hypothesis sentences. Several submissions, including the top two scoring systems, used the C&C Parser (Clark and Curran 2007) which is based on the

Combinatory Categorical Grammar (CCG; Steedman (2000)) formalism. Others used dependency structures produced by MaltParser (Nivre et al 2007b), MSTParser (McDonald et al 2005) and Stanford Parser (Klein and Manning 2003).

After the parsing step, the decision for the entailment was based on the comparison of relations, predicates, or dependency paths between the text and the hypothesis. Most systems relied on heuristic methods of comparison. A notable exception is the MARS-3 system which used an SVM-based classifier to decide on the entailment using dependency path features.

The top two scoring systems, Cambridge and SCHWA (University of Sydney), were based on the c&c parser and used a similar approach (though Cambridge used gr output in SD format while SCHWA used the native ccg dependency output of the parser). They achieved almost identical task accuracies, but SCHWA was more accurate on "yes" entailments, while Cambridge was more accurate on "no" entailments, resulting in a higher overall accuracy for Cambridge, but a higher F-score on positive entailments for SCHWA (Table 4). We attribute this difference to the decision criteria used in the entailment systems, which will be discussed in Section 4, but notably the difference suggests that a dependency-based entailment system can be tuned to favour precision or recall.

The following sections describe the Cambridge system in more detail and present detailed error analyses for the parser and the entailment system.

4 The Cambridge System

The best–scoring Cambridge system used the c&c parser (Clark and Curran 2007), which can produce gr output in SD format (see Section 1) using custom tools available with the parser. [4]

The entailment system was very simple, and based on the assumption that H is a simplified version of T, which is true for this task though not necessarily for RTE in general. Let grs(S) be the grs produced by the parser for a sentence S. The basic intuition is that if grs(H) $\subseteq$ grs(T), then in principle H should be considered an entailment of T. In practice, certain refinements of this basic intuition were required to account for non-matching grs resulting from grammatical transformations used in entailment construction, or noise in the parse which could be safely ignored.

Three situations were identified in which grs in H would not exactly match those in T. First, syntactic transformations used in entailment construction could change head-dependent relations. By far the most frequently used transformation in the PETE dataset was passivization.

4 http://svn.ask.it.usyd.edu.au/trac/candc

tailment construction meant that H could contain tokens not present in T. Second, the introduction of dummy words and transformations during en-
This included pronouns such as “somebody” and “something”, auxiliary verbs introduced by passivization, and expletive subjects. In addition, determiners were sometimes introduced or changed, eg “prices” to “the prices”.

Third, the parses of T and H might be inconsistent in a way incidental to the target entailment. Cümle çifti T düşünün: Ben uzandı benim elbise kadar yüksek olduğu küçük komik cep. ⇒ H: Cep kadar yüksek şey. Değerlendirmenin amacı odağı arasındaki ilişki ise "Cep" ve "yüksek". Sürece ayrıştırıcı konu olarak "cep" analizleri ile "yüksek" bir, biz PP başlangıcı bağlama, diyelim ki, bunu cezalandırmak önlemek istiyorsanız ile "yukarı" farklı T ve H.

Bu sorunları çözmek için sistem sezgisel küçük bir set kullanılır. İlk olarak, o Bu ele T. değil bir belirteç içeren gr'lık herhangi gr (H) göz ardı zamirler, pasif yardımcılar, küfür konular ve belirleyicileri. İkincisi, o Doğrudan nesneler ile eşdeğer pasif konular. Benzer sezgiseller tanımlanabilir Diğer dönüşümleri karşılamak için, sadece bu uygulanmıştır, kalkınma setinin incelenmesine dayanan.

Üçüncü olarak, kontrol ederken gr (H) ⊆ gr (T), sadece çekirdek ilişkileri subfertilitesi olsun JECT ve nesne olarak kabul edildi. Niyet tesadüfi farklılıklar oldu T ve H ayrıştırır arasındaki hataları olarak sayılmayacaktır olacaktır. Bu gr türleri kalkınma kümesindeki entailments incelenmesine dayanan seçildi, ancak sistem kolayca diğer ilişki türlerine odaklanmak için yeniden olabilir, PP-eki görev için örneğin PP ilişkileri.

Son olarak, (H) ∩ gr (T) sistem gerekli gr olmayan boş (anlamsız olduğu Pozitif), ancak konular ve nesnelere bu kriter kısıtlamak değil.

Sistem bir PTB dizgeciklerini kullanır[5] tümleyici Train- ile tutarlılık için veri ing. Içine yerleştirilmiş Morpha lemmatizer (Minnen ve ark, 2000), c c araçları, T ve H genelinde belirteçleri eşleştirmek için kullanılan ve tüm simgeleri kon- idi küçük harfe invert. Ayrıştırıcı T ya bir yayılan analiz bulmak için başarısız olursa veya H, gerektirim kararın "hayır" oldu. Tam boru hattı, Şekil 1 'de gösterilmiştir.

Schwa sistemi, Cambridge Cambridge sistemini karşılaştıran "evet" entailments üzerinde "hayır" entailments ve Schwa daha doğru oldu. We Her iki sistem, en az bir eşleme ilişkisi gerekli çünkü bu olduğuna inanıyorum T ve "evet" cevabını H, fakat Cambridge arasında ek cevap "Hayır" (özne veya nesne) herhangi bir çekirdek ilişkisi T. değilse H mevcuttu ama Böylece Cambridge Schwa daha az yanlış pozitif izin verdi.

5 Hata Analizi

Tablo 4 test seti Cambridge sistemine ait sonuçlar içermektedir. On the geliştirme sistemi% 66,7 genel bir doğruluk elde ayarlayın. Olumlu entailments, hassas hatırlama 0,5789 oldu 0,7857 oldu, ve F-skoru 0,6666 oldu.

5 http://www.cis.upenn.edu/~treebank/tokenizer.sed

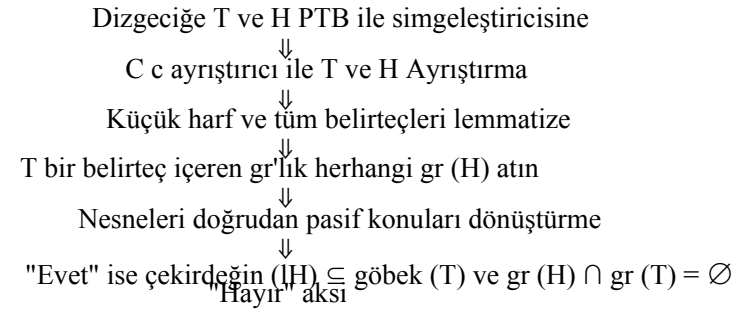

Fig. C c ayrıştırıcı ve Vasiyetiniz sistem için 1 tam boru hattı. Çekirdek (S): çekirdek seti (konu ve nesne) gr gr'lık (S) in.

Tablo 5 gelişmekte çeşitli gramer ilişkilerin sıklığını listeler Cambridge sistem hatalar yaptık opment ve test seti örnekleri. A Tablo 3 ile karşılaştırılması doğrudan nesneleri ve azaltılmış göreli hükümler göstermektedir Hatanın sık sebepleridir.

GR	Dev Seti	Deney Seti
İndirimli göreli fıkra	45%	36%
Doğrudan nesne	18%	51%
Bağıl fıkra	18%	—
Nominal konu	9%	20%
Pasif Nominal konu	9%	7%
Zarf değiştirici	9%	—
Edat değiştirici	9%	—
Bitişik	9%	—
Edat nesne	—	7%

Yanlış cevaplar var Vasiyetiniz durumlarda gramer ilişkiler Tablo 5 Frekans Cambridge sisteminden.

Gösterecek şekilde ayarlanmış geliştirme sonuçlarının aşağı Tablo 6 ileri sonları ne kadar farklı ayrıştırıcı türleri ve Vasiyetiniz sistem hataları katkıda Yanlış cevaplar. Vakaların çoğunda ayrıştırıcı ve Vasiyetiniz sistemi beklendiği gibi doğru cevabı bulmak için birlikte çalıştı. Örneğin, T: AMR hisse senetleri Ticaret EDT Cuma 3 pm kısa bir süre sonra askıya alınmıştı ve devam etmedi. H ⇒. Ticaret devam etmedi, çözümleyici üç gr dan üretilen H (belirteçleri lemmatized ve küçük gösterilmiştir): (ticaret devam nsubj), (neg yok) ve (do aux özgeçmiş). Bunların hepsi gr'lık (T) de yer aldı ve Doğru "evet" kararı alındı. T: Moreland tam için kara kara oturdu dakika, bu süre boyunca ben her birimiz yeni bir içki yaptı. H ⇒: Dakika yapılır. H. One ayrıştırıcı üretilen iki gr (auxpass olmak yapmak), göz ardı edildi Pasif yardımcı "olmak" çünkü T. ikinci, pasif konu değildir gr (nsubjpass dakika yapmak) dobj yapmak (doğrudan nesne ile eşit oldu dakika). Bu gr gr'lık (T) değildi, bu yüzden doğru "hayır" kararı alındı.

Bazı durumlarda doğru bir "evet" cevabı tartışmasız yetersiz yoluyla ulaşıldı Pozitif kanıtlar. T: O gece perde ortasında uyanmak bu konuda. H ⇒. O uyanmak, çözümleyici için yanlış analizler üretiyor T ve H. hem Ancak, bu gr göz ardı edilir için VP "uyanmak" onlar non-core (özne veya nesne değil), ve "evet" kararı dayalı olduğu için Tek gr maç (nsubj o olur). Bu tamamen şanslı bir tahmin değil, Vasiyetiniz sistemi doğru hatalı analizler göz ardı beri "would uyanmasını "ve rolü üzerinde duruldu" dedi "cümlenin öznesi olarak. Ancak hedef konu "dedi," arasındaki ilişki, özellikle beri ve sözcük fiil "uyanıklık", daha olumlu bir kanıt istenebilecektir. Of the 22 Doğru "evet" kararlar, sadece iki o tek gerçekten şanslı tahmin edildi maç belirleyici oldu; tüm diğerleri en azından bir çekirdek maç vardı.

Type	FN	FP	Total
Sınırsız bağımlılık	9	1	10
Diğer ayrıştırıcı hatası	6	2	8
Entailment sistemi	1	3	4
Total	16	6	22

Kalkınma sette Tablo 6 Hata dökümü. FN: yalancı negatiflik, FP: yalancı pozitif.

Tablo 6 hataları dökümünü gösterir. En büyük kategori yanlış oldu sınırsız bağımlılıkları nedeniyle negatifleri için, çözümleyici tarafından kurtarıldı değil örnek T: O artık hakkında satirik olmak için sahip bir enerji gereklidir onun babası. H ⇒: Biri artık enerjiyi sahipti. İşte ayrıştırıcı başarısız T. "sahip" ve "enerji" Öyle arasında doğrudan nesne ilişkisini kurtarmak için ayrıştırıcıları sınırsız bağımlılıkları ile zorluk bilinen ki (Rimell ark 2009), bu yüzden, bu sonuç şaşırtıcı değildir. Sınırsız bağımlılıkları arasında, o içerdiğinden, bir altın standart Vasiyetiniz özellikle zordu Ekstraksiyon iki kat; Bu T idi: Index-arbitraj ticaret "bir şeydir yakından izlemek istiyorsanız, "Londra'daki Menkul Kıymetler Borsası resmi bir söyledi. ⇒ H: Biz index-arbitraj ticareti izlemek istiyorum. Burada, gerektirim kurtarma gerektirir, sadece doğru "bir şey" başkanlığında göreceli fıkra ayrıştırma, aynı zamanda "ticaret" olarak "bir şey", bir bileşik görev başvurusu çözme Bu birkaç modern sözdizimsel ayrıştırıcıları girişimi. Yine de, bu bir meşru Vasiyetiniz, Bölüm 2.2 de kurallarına göre inşa edilmiştir.

Bir sonraki kategori, diğer ayrıştırıcı hataları oldu. Bu çeşitli kategori örneğin koordinasyon hatalar, parantez unsurları, belirlenmesi de dahil olmak üzere Bir Clausal konunun başkanı ve POS tagger nedeniyle bir hata. For example, T: O zaman en azından o onun araçları ve bir şeyler asmak için bir yer olurdu üzerinde çalışmak. H ⇒: O üzerinde çalışmak için bir şey olurdu, çözümleyici yanlış. koordineli "araçlar" ve T "bir şey", "bir şey" yapma gibi görünen "asılı" bir amacı. Bunun bir sonucu olarak (dobj şey) (H) gr'lık oldu ama değil gr (T), yanlış bir "hayır" verimli.

Dört hataları yerine ayrıştırıcı daha Vasiyetiniz sistemi nedeniyle vardı; these Bölüm 6'da dicsussed edilecektir.

Page 16

6 entailment Sistemi Değerlendirmesi

Bir ayrıştırıcı değerlendirme aracı olarak PETE faydası de bulunmas bağlıdır
Uygun Vasiyetiniz sistemlerinin lık, gerektirim sistemi gibi davranır çünkü
ayrıştırıcı ve PETE veri kümesi arasındaki aracı. Biz inanıyoruz AP-
Bir Vasiyetiniz sisteminin propriate rolü açısından tamamen şeffaf olmak
olan ayrıştırıcı çıkışı için, sadakatle ayrıştırıcı yapıp yapmadığını yansıtmak için
eki H "evet" cevabı gerektirir T karar (lar), ne içi
troducing ne de herhangi yanlışlarını düzeltme. Örnek T hatırlatarak: man ile
şapka ayrıştırıcı "yorgun" bir konu olarak "adam" analizleri ise, yorgun,
Vasiyetiniz sistemi H-1 için "evet" cevabını boyunca geçmelidir: adamdı
H-2 yorgun ve bir "hayır" cevabı: şapka yorgundu. Vasiyetiniz sistemiyse
Bu şekilde, şeffaf, daha sonra H üzerinde bir doğru cevap bir doğru ayrıştırma gösterir
ve H yanlış cevap (H içeriği bakımından) gösteren bir
ayrıştırıcı hatası.

Bu saydam olmayan Vasiyetiniz sistemleri çeşitli hayal etmek kolaydır. For
Her zaman ayrıştırıcı olursa olsun "evet" cevapları örneği, bir gerektirim sistemi
"Her zaman evet" başlangıca tam olarak eşit bir doğruluk puanı verecek çıkışı,
olursa olsun ne kadar iyi ya da kötü yatan ayrıştırıcı yürüttü. On the
Öte yandan, çözümleyici çıkış geçersiz kılmak için izin verilen bir Vasiyetiniz sistemi
Arka plan bilgisi veya akıl buluşsal dayalı da olmayan olurdu
Şeffaf o çözümleyici kararları artırabilirsiniz çünkü. Biz değil Not
Böyle bir Vasiyetiniz sistem için NLP hiçbir rolü, yalnızca o olur, olduğunu söylemek
şeffaf PETE veri kümesi üzerinde ayrıştırıcı doğruluğu iletmek değil. Moreover,
gerektirme bir ara ürün, yarı şeffaf görünüme sahiptir ilişkin olabilir
çözümleyici tarafından yapılan ekleri kılamaz, ancak ekleyebilir sistemi
information.

O Aracı olabildiğince ise biz PETE görev geçerli bir ayrıştırıcı değerlendirme aracı olduğunu söyleyebiliriz
herhangi bir çözümleyici için uygun bir Vasiyetiniz sistemi inşa etmek ble. We
Tüm katılımcı ve ister Vasiyetiniz sistemlerini değerlendirmek çalışmayın
mesi sistemleri, uygun, ama bir vaka çalışması olarak, biz olmadığını düşünün
Cambridge Vasiyetiniz sistemi değerlendirmek için uygun bir araç oldu
PETE veri kümesi c & c ayrıştırıcı. Daha genel Vasiyetiniz sistemidir
Bölüm 7'de tarif.

Biz yalıtmak ve performansını değerlendirmek için iki torpil deneyleri kullanın
Cambridge Vasiyetiniz sistemi. İlk kehanet deneyi altın kullanır
Standart gr ziyade gerektirme girdi olarak otomatik ayrıştırıcı çıkışı
system. Cambridge gr-temelli yaklaşım geçerli olduğunu varsayarak, daha sonra verilen altın
T ve H için standart gr, biz uygun bir Vasiyetiniz sistemi bekliyoruz
Tüm ayrıştırır doğru olduğundan (görev değerlendirmede% 100 doğruluk sonucu,
ve Vasiyetiniz sistemi sadakatle) Doğru analizler boyunca geçmelidir.
Bu deneyi gerçekleştirmek için elle tüm T ve H cümleler açıklamalı
kalkınma altın standart gr'lık ile ayarlayın. Cambridge Vasiyetiniz kullanma
altın gr'lık sistem% 90.9 bir görev doğruluğu puanı, sonuçlandı
Bu nedenle, aynı zamanda bu Vasiyetiniz sistemi doğruluk skoru olarak kabul edilebilir
experiment.

Page 17

Sistem tarafından yapılan altı hataların, üç (iki FN ve bir FP) idi
nedeniyle gr etiket veya baş değiştirdi T ve H arasındaki dönüşümlere.
Örneğin, T düşünün: Bazen çocuklar buğulanmış bulmak, tam buğday
Onlar "buckshot" dediğimiz tahıl taneleri için. H ⇒: Tahıllar buharda. T olarak,
"Atmak" onun başkanı olarak "tahıl" ile prenominal sıfattır; H ise, o
onun konu olarak "tahıl" ile bir pasif. Vasiyetiniz sistemi hesap vermedi
Bu dönüşüm için. Prensip olarak, herhangi bir dönüştürme için sorumlu olabilir
bu da oluşturmaktadır, aynı şekilde gr'lık bir kural olarak ifade edilebilir
edilgenleştirme. Uygulamada, tek yol Vasiyetiniz sistemi garanti
veri kümesi içinde T ve H arasındaki tüm dönüşümler için hesap için
olası dönüşümlerin tam listesi gerektirme belgelenmelidir
nesil kurallar ve geliştiriciler için sunulan.

İki hataları (her ikisi de FP) meydana geldiğinde bir çekirdek olmayan ilişkisini içeren gr veya
Sistem gözardı her ikisi de H tanıtılan bir zamir, hayati bir önem taşıyan
Doğru Vasiyetiniz kararı. Olmayan görmezden geliştirme kararı yana
H tanıtıldı çekirdek ilişkiler ve zamirler, genel doğruluk için yapıldı
hatalar bu şartlar altında kaçınılmaz, ancak küçük bir yüzdesi vardır
veri kümesi içinde cümle.

Nihai hata tartışılan zor sınırsız bağımlılık oldu
Bölüm 5, T: Index-arbitraj ticaret ", bir şey biz yakından izlemek istiyorum" dir
Londra'nın Menkul Kıymetler Borsasında resmi söyledi. H ⇒: Biz endeksinde izlemek istiyorum
arbitraj ticareti. Altın gr, doğru sınırsız bağımlılık temsil
ama yine de rEF- çözmek için gerekli bilgileri vermeyin
"ticaret" olarak "bir şey" nin ferans.

İkinci deneme Cambridge Vasiyetiniz sistemi karşılaştırıldı
Bir kehanet Vasiyetiniz sistemi, T, H gerektirir konusunda manuel yargılar yani
çözümleyici analiz verilen. Biz otomatik c & c tarafından oluşturulan gr dan kullanılan
kalkınma kümesi için, ve elle her cümle için karar T olsun
gerektirdiği H sadece otomatik ayrıştırıcı analizine dayalı. Biz o zaman bu karşılaştırıldı
Otomatik Vasiyetiniz sistem kararları ile manuel analiz belirlemek için
nasıl şeffaf Vasiyetiniz sistemi oldu.

Manuel analizlere dayanarak, biz Cambridge gerektirim sistemi bulundu
Otomatik olarak oluşturulan kullanarak geliştirme seti, altı hataları yapılmış
c & c çıktı. Ayrıştırıcı analizi oldu yani iki hata, çözümleyici yanaydı
Yanlış, ancak gerektirim sistem hatası "düzeltilmiş"; dört edildi
ayrıştırıcı analizi yani kendi zararına, doğru ama Vasiyetiniz kararı
Yanlış oldu. Vasiyetiniz sisteminin doğruluğu, bu ölçümde 90.9% idi
(ER- olsa önceki kehanet deney sonuçları ile uyumludur
roneous cümleler) aynı değildi.

Çözümleyici lehine bir yer T iki hatalar arasında: Onlar istedi
Genel takımından - sırtını gibi hissettim ne olduğunu görmek için. H ⇒: Biri görmek istedim
Ne sırtını gibi hissettim. Ayrıştırıcı sırtını hissettim "ifadesini analiz
gibi "yanlış ama, T ve H ikisi için aynı hata yapılmış, böylece gr
eşleşti. Sadece manuel analizi ile hata bulunamadı. Diğer hata
eki kararın yanlış bir cümle oldu ama oldu
Bir belirleyici tek gr maç.

Page 18

Ayrıştırıcı koreleydi zararına dört Vasiyetiniz sistem hataları
sını Tablo "entailment sistemi" aralıksız ayrıca bu hatalardan 6. Three
İlk oracle deneyde meydana gelen, yukarıda tartışılmıştır. The
Dördüncü T, T ve H arasındaki POS değişiklikten kaynaklanan: Orada
Biz olmasaydı davrandı Tibet'te devrim. ⇒ H: yapmacık
yoktu. "Hayır" cevabını bulmak için hayati gr oldu (nsubj var
gr'lık (H)) gibi davrandı, ama gerektirim sistem lem- çünkü göz ardı
matizer bir isim olarak "sahte" için lemma olarak "taklit" vermedi. This
Hata türü "hayır" ise cevaplayarak önlenebilir olabilecek kelimenin POS
Uygulama Önemsiz olmayan olacağını, ancak T ve H arasındaki değişimler,
Kelime endeksleri de değişebilir çünkü.[6]

Not Vasiyetiniz sistem için% 90.9 doğruluk rakam dayanan
İkinci kehanet deneyde manuel analiz ayrıştırıcı yansıtmamaktadır
doğruluk: çözümleyici önemli bir bağımlılık bir hata yaptığınız zaman, lider
Yanlış bir Vasiyetiniz karar, biz olmak Vasiyetiniz sistemi değerlendirilecektir
sadakatle amacıyla çözümleyici hata birlikte geçti beri, düzeltin
ayrıştırıcı değerlendirilmesi. C c ayrıştırıcı bir kahin ile birleştiğinde olsaydı (örneğin
tamamen manuel) Vasiyetiniz sistemi, bu% 69.7 doğruluk üzerine elde olurdu
otomatik ile elde% 66.7 oranla gelişme seti,
Vasiyetiniz sistemi.

Basit Cambridge Vasiyetiniz sistemi için yüksek doğruluk seviyeleri
c performansı boyunca geçen at% 90'ın üzerinde doğru olan ve üzerinde c
PETE veri seti, oldukça ümit vericidir. Bazı ek doğruluk olabilir
ayrıştırıcı ve Vasiyetiniz sistem geliştiricilerin ac- varsa gelecekte kurtarıldı
dilsel dönüşümün geliştirilmiş belgelere cess izin
Vasiyetiniz nesil. Bu PETE geçerliliği için olumlu bir sonuçtur
task.

7 A Genelleştirilmiş entailment Sistemi

Bu görev üzerinde daha fazla araştırma kolaylaştırmak için, bir diğer uygulamaya
Cambridge sistemine dayalı modüler ve genelleştirilmiş Vasiyetiniz sistemi.
Benzer sezgiselleri ve arama yöntemleri uygulayarak, biz çoğaltmak mümkün
Üst görev skor, hem de farklı ayrıştırma paradigmaları karşılaştırmak mümkün
Daha düzey oyun alanı için yapım, tek bir gerektirim sistemi kullanılarak.

Bu sistem ile entegre dolayısıyla girdi olarak Stanford bağımlılıkları alır ve
Stanford üretebilen çok sayıda halka açık ayrıştırıcıları
Yazılan bağımlılıklar. Kullanılan ayrıştırıcıları Berkeley Ayrıştırıcı (Petrov ve vardı
Klein 2007), Charniak Ayrıştırıcı (Charniak ve 2005 Johnson), Collins Ayrıştırıcı
(Collins 2003), c c Ayrıştırıcı (Clark ve 2007 Curran), Malt Ayrıştırıcı (Nivre ark

6 Ile sonuçlanmayan çoğu gelişimi sette sekiz POS değişiklik vardı
değerlendirmede hatalar. Bu özel H kuralsız İngilizce olduğunu da unutmayın. Hatırlamak
Negatif H cümleler hakiki ayrıştırıcı hataları elde edilmiştir; her zaman mümkün değildi
Biz ele alacağız rağmen, bu tür hatalara karşılık gramer cümleler inşa etmek
Tüm H cümleler sınırlayıcı gelecekteki çalışma dilbilgisi olmak.

Page 19

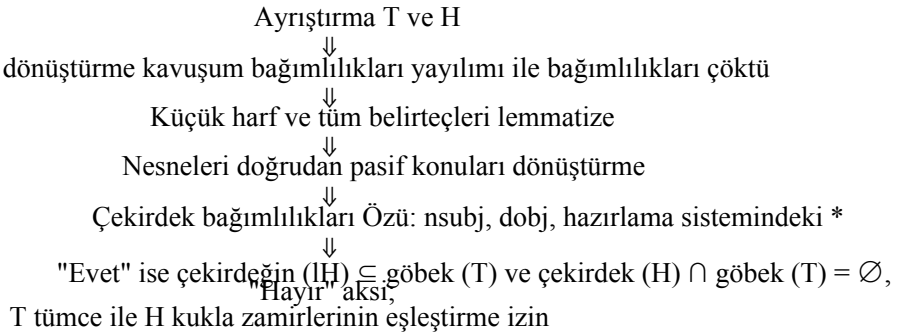

Fig. Stanford için 2 tam boru hattı Vasiyetiniz sistemi bağımlılıkları. Çekirdek (S): çekirdek grubu (özne, nesne ve edat) gr'lık (S) 'de GRS.

2007b), MSTParser (McDonald ve ark 2005) ve Stanford Ayrıştırıcı (Klein ve) 2003 Manning. Her ayrıştırıcı WSJ bölümünün bölümlerinde 02-21 eğitilmiş edildi Penn Treebank evi. MaltParser ve MSTParser Stanford eğitilmiştir Penn Treebank bağımlılıkları biçimi (Cer ve diğerleri 2010) 'de tarif edildiği gibi c c CCGbank (Hockenmaier 2003) eğitim gördü.

Stanford bağımlılıkları birçok çeşitleri (De Marneffe ve gelip yazdığınız) 2008 Manning. Cambridge sistem ağacı kıran de- seçeneği kullanılır pendency tipleri, kavuşum bağımlılıklar, xsubj ve pObj ve yayılımını ref ama hiçbir bağımlılıkları çöktü. Stan- kullanılan genelleştirilmiş Vasiyetiniz sistemi ford ağaç kırma dahil tam çökmüş bağımlılıkları ile (bağımlılıkları bağımlılıklar) ve kavuşum bağımlılıkları yayılması. Tam col- kullanma lapsed bağımlılıklar çeşitli re- geliştirilmiş eşleştirme izin düşünülmüştü T ve H. arasındaki saplamalar, özellikle edatlar,

C & c dışındaki tüm ayrıştırıcıları çıkışları Stanford col- dönüştürüldü kavuşum bağımlılıkları yayılması ile lapsed bağımlılık gösterimi Stanford Ayrıştırıcı kullanarak. C c ayrıştırıcı bir, birbirlerinden üretmek değil, çünkü Stanford araçları ile dönüşüm için uygun resentation biz dönüştürülmüş Stanford c & c çıkış Bölüm 4'te olduğu gibi, özel araçlarını kullanarak bağımlılıkları.

Entailments iki test ve hipotez cümlelerin üzerine karar vermek olduğunu çözümlü. T ve H tüm kelimeler lemmatized ve pars- sonra küçük harfli edildi ing. Biz aktif pasif Cambridge sisteminde aynı sezgisel geçerlidir Dönüşüm ve kukla kelime eşleştirme. Daha sonra çekirdek bağımlılık dikkate tipleri, dobj nsubj ve hazırlama sistemindeki * T ve H (col- kullanımını karşılaştırırken lapsed bağımlılık temsil edatları dikkate mümkün kılar) bir çekirdek ilişkisi olarak. En az bir çekirdek H bağımlılık ve tüm çekirdek varsa H bağımlılıkları da T bulunur, kararın "evet" tir. Çekirdek H de- Eğer pendencies kararının "hayır" dır T bulunmaz. Tam boru hattı gösterilir Şekil 2'de.

Tablo 7 elde edilen sonuçları listeler. Önemli farklılıklar vardır karşılaştırılabilir etiketsiz eki olan sistemlerin Vasiyetiniz doğrulukları CoNLL ayrıştırıcı çıktısından elde UAS ile puanları (UAS), temsil

rin sunulması. Bu farkın potansiyel bir nedeni bileşimi Zorlu sözdizimsel yapılar, bazı vurgular PETE kümesi ayrıştırıcıları daha iyi olabilir. UAS ve PETE puanları arasındaki fark yeniden anlamsal için UAS gibi Treebank kayıtsızlığı bazlı önlemler flects Çeşitli bağımlılıkları önemi ve potansiyel uygulamalar üzerindeki etkisi.

Stanford bağımlılıkları yolda farklı dönüşüm adımları Not may Sonuçlar tüm ayrıştırıcıları için tam karşılaştırılabilir olmadığı anlamına gelir. Nevertheless, the PETE puanı ayrıştırıcı sonuçlar üzerinde özellikle farklı bir bakış açısı sağlar. We İki bağımlılık ayrıştırıcılar, MaltParser ve MSTParser, düşük değerli göstermektedir dikkat kurucu ayrıştırıcıları daha doğrulukları; Bu sonuçlar ile tutarlıdır (Cer ve arkadaşları 2010). Hiçbir UAS CoNLL üretmek değil c & c için kullanılabilir tarzı çıktı.

System	PETE	UAS
C & C Ayrıştırıcı	73,42%	–
Collins Ayrıştırıcı	71.43%	91.6
Berkeley Ayrıştırıcı	71,10%	91.2
Charniak Ayrıştırıcı	68,44%	93.2
Stanford Ayrıştırıcı	67.11%	90.2
MaltParser	64,12%	89.8
MSTParser	62,46%	92.0

p-value	Coll	Berk	Char	Stan	Malt	MST
C & C Ayrıştırıcı	0,5663	0,4567	0,0966	0,0351	0,0032	0,0004
Collins Ayrıştırıcı	1.0		0,2893	0,1056	0,0108	0,0012
Berkeley Ayrıştırıcı			0,2299	0,0667	0,0192	0,0024
Charniak Ayrıştırıcı				0,6582	0,1485	0,0421
Stanford Ayrıştırıcı					0,3134	0,1149
MaltParser						0,5595

Tablo 7 Örnek sistemleri: ilk tablo PETE test kümesi üzerinde performans verir ve Penn Treebank bölüm 23 etiketsiz bağlanma puan verin. İkinci tablo verir McNemar testine göre PETE puanları arasındaki farklar için p-değerleri (Dietterich 1998). Istatistiksel olarak anlamlı fark (p <.05) kalın yazı ile belirtilmiştir.

8 tarihinden

Biz metinsel tr kullanarak Pete, çözümleyici değerlendirilmesi için yeni bir yöntem tanıttı tailments. Bağımlılıkları entailments dayandırarak o mevcut durumu sanat ayrıştırıcıları bu olur biz bir veri kümesi oluşturmak için umut, üzerinde hata yaparlar

ortak değerlendirme ölçümleri kullanarak dikkat. Ayrıca kendimizi kısıtlayarak
doğal dil entailments tarafından ifade edilebilir farklılıklar, biz ümit
semantik ilgili kararları yerine sözleşmenin kazaları odaklanmak
hangi ortak değerlendirme ölçümlerini kadar karışık olsun. Biz olmayan güvenmeye seçti
Doğal çıkarım görev eğitilmiş annotators ziyade eğitimli annotators
yapay etiketleme görev inanıyoruz, çünkü (i) com- birçok alt alanlar
yalnızca sayısal dilbilim gürültü nedeniyle içinde ilerleme yapmak için mücadele

yapay verileri etiketli, ve (ii) sistemleri, doğal Komisyonu modellemek için çalışmalısınız
Onların kusurlu yerine cümleleri anlamakta annotators arasında yetkinlik
Yapay etiketleme görevleri performans.

Bölüm 7'de tarif edilen örnekler dahil olmak üzere çoklu sistemler, elde
state-of-the-art ayrıştırıcıları ve basit en- kullanarak PETE görev iyi sonuçlar
tailment sistemleri. Cambridge Vasiyetiniz sisteminin analizi gösterdi
bu c & c değerlendirmek için bir araç olarak yaklaşık% 90 doğruluk var
ayrıştırıcı, ya da potansiyel PETE geliştirme gr dan üreten herhangi bir ayrıştırıcı,
data. Bu sonuç belki de bu yana görev puanları daha da önemli olduğunu
PETE bir ayrıştırıcı değerlendirme yaklaşımı olarak takip değer olduğunu göstermektedir.

Umudumuz PETE gibi veri setleri değerlendirme için değil, aynı zamanda sadece kullanılacak olan
Eğitim ve gelecekte sistemlerinin ince ayar için. Daha fazla iş için gerekli olan
Vasiyetiniz nesil işlemi otomatik ve kompozisyon dengelemek için
Bir PETE veri kümesi kaplı sözdizimsel olayların.

Acknowledgments

Biz onların dikkatli analiz için Stephan Oepen ve Anna Mac teşekkür etmek istiyorum
ve değerli suggestions.Onder Eker Zehra Turgut de- katkıda
PETE görev velopment. Stephen Clark geliştirme işbirliği
Cambridge sisteminin. Biz de için Matthew Honnibal teşekkür etmek istiyorum
Vasiyetiniz sistemine Schwa sistemi ve katkısının tartışılması
analysis.

References

Siyah E, Abney S Flickenger D Gdaniec C Grishman R, Harrison P Hindle D Ingria R, Jelinek F, Klavans J, ve diğerleri (1991) Kantitatif syn- karşılaştırılması için bir Yöntem İngilizce Dilbilgilerinin taktik Kapsamı. In: Konuşma ve dilin doğal: tutanaklarının Pacific Grove, Kaliforniya, Şubat 19-22, 1991, Morgan Kaufmann düzenlenen bir atölye, Birahane, s 306

Bonnema R, Bod R, scha R (1997) anlamsal yorumlanması için bir DOP modeli. In: davası Hesaplamalı Derneği'nin Avrupa bölüm sekizinci konferansın rinde

Bos J, Hesapla (eds) (2008) Çapraz-framework Dilbilim Çalıştayı Bildiriler Kitabı
Kon- Domain Ayrıştırıcı Değerlendirilmesi, URL http://lingo.stanford.edu/events/08/pe/,
Hesaplamalı Dilbilim 22. Uluslararası Konferansı ile Nection
Carroll J, Minnen G, Briscoe T (1999) çözümleyici değerlendirme için Corpus açıklama. In: Pro-
Dilsel yorumlanır Corpora üzerinde EACL çalıştayın bu davalar (Linc)
Cer D, Marneffe MC, Jurafsky D, Manning CD (2010) de stanford dependen- için Ayrıştırma
leri: hız ve doğruluk arasındaki ticaret-off. In: Diller Yüksekokulu 7. Uluslararası Konferansı
guage Kaynakları ve Değerlendirme (2010 LREC), URL http://nlp.stanford.edu/pubs/
lrecstanforddeps \ _final \ _final.pdf
Charniak E, Johnson M (2005) Kaba-to-ince n-iyi ayrıştırma ve ayırt edici MaxEnt
yeniden yükselmesini sağlamıştır. In: Hesaplamalı Derneği ile ilgili 43 Yıllık Toplantısı Proceedings
Bilişimsel Dilbilim için Dilbilim Derneği, s 180
Clark S, Curran J (2007) Geniş kapsama verimli istatistiksel ccg ile ayrıştırma ve log-lineer
models. Dilbilim 33 (4): 493-552

Page 22

Collins M (2003) doğal dil ayrıştırma Başkanı odaklı istatistiksel modeller. Hesaplamalı
dil 29 (4): 589-637
Dagan I Dolan B, MAGNINI B, Roth D (2009) tanıma metinsel Vasiyetiniz: Akılcı,
değerlendirme ve yaklaşımlar. Doğal Dil Mühendisliği 15 (04)
De Marneffe M, Manning C (2008) Stanford bağımlılıkları kılavuzu yazdınız. URL http: // nlp.
stanford.edu/software/dependencies-manual.pdf
De Marneffe M, MacCartney B, C Manning (2006) oluşturuluyor yazdığınız bağımlılık ayrıştırır
ifade den yapı ayrıştırır. In: LREC 2006
Dickinson M, MEURERS WD (2003) treebanks içinde algılama tutarsızlıklar. In: Pro-
Treebanks ve Dil Teorileri İkinci Çalıştayı (TLT ve bu davalar
2003), Växjö, İsveç, s 45-56, URL \ url {http://ling.osu.edu/dickinso/papers/
Dickinson-MEURERS-tlt03.html}
Denetlenen sınıflandırma karşılaştırmak için Dietterich T (1998) Yaklaşık istatistiksel testler
öğrenme algoritmaları. Sinirsel hesaplama 10 (7): 1895-1923
Erk K, McCarthy D, Gaylord N (2009) kelime duyular ve kelime kullanımları üzerinde araştırmalar. In:
ACL ve 4. 47 Yıllık Toplantısı Ortak Konferansı Bildiriler Kitabı
AFNLP Doğal Dil İşleme Uluslararası Ortak Konferansı: Hacim
1-Cilt 1, Hesaplamalı Dilbilim Derneği, s 10-18
Hockenmaier J (2003) Veri ve kombinatoriyel kategorik istatistiksel ayrıştırma için modeller
dilbilgisi. Edinburgh Doktora tezi, University of
Hockenmaier J, Steedman M (2007) CCGbank: CCG sözcükler ve bağımlılık derlem
Penn Treebank çıkarılan yapılar. Dilbilim 33 (3): 355-396
Kral T, Crouch R, S Riezler, Dalrymple M, Kaplan R (2003) PARC 700 bağımlılık
bank. In: EACL03 Bildiriler Kitabı: dilbilimsel 4. Uluslararası Çalıştayı
Yorumlanır Corpora (LINC-03), sayfa 1-8
Klein D, Manning C (2003) Doğru görünüm bilgisi ayrıştırma. In: 41. Proceedings
Hesaplamalı Dilbilim-Cilt 1, Dernek Derneği Yıllık Toplantısı ile ilgili
İşlemsel Dilbilimleri, s 423-430
Marcus M, Santorini B, Marcinkiewicz M (1994) En- büyük açıklamalı külliyat Bina
Türk çe: Penn Treebank. Bilişimsel dilbilim 19 (2): 313-330
McCarthy D, Navigli R (2007) Semeval 2007 görev 10: İngilizce sözcük değiştirme görevi. In:
Semantik Değerlendirmeleri Dördüncü Uluslararası Çalıştayı (SemEval- Bildiriler Kitabı
2007), Hesaplamalı Dilbilim, Prag, Çek Cumhuriyeti, pp 48-53 Derneği,
URL http://www.aclweb.org/anthology/W/W07/W07-2009
McDonald R, Pereira F, Ribarov K, Hajic J (2005) Sigara yansıtmalı bağımlılık kullanarak ayrıştırma
Ağaç algoritmaları kapsayan. In: HLT Proceedings / EMNLP, s 523-530
Minnen G, Carroll J, Pearce D Sağlam (2000), morfolojik nesil uyguladı. In: Pro-
INLG, Mitzpe Ramon, İsrail bu davalar

2007 Salomi L Kubler S Marinov S Marsi E Nilsson J Nivre J Yuret D (2007a) CoNLL
Nivre J, Hall J, Kubler S, McDonald R, Nilsson J, Riedel S, Yuret D (2007a) The CoNLL
EMNLP-CoNLL, cilt 7, sayfa 915-932 arasında Bölüm
Nivre J, Salon J, Nilsson J, Chanev A, Eryiğit G, Kubler S, S Marinov, Marsi E (2007b)
MaltParser: veri tabanlı bağımlılık ayrıştırma için bir dilden bağımsız sistem. Nat-
ural dil Mühendisliği 13 (02): 95-135
Petrov, S, D Klein (2007), görünüm bilgisi ayrıştırma için geliştirilmiş çıkarsama. In: Kitabı
NAACL HLT 2007, sayfa 404-411
Ayrıştırıcı değerlendirmesine için Rimell L Clark S, M Steedman (2009) Sınırsız bağımlılık kurtarma
tion. In: Doğal Dil Ampirik Yöntemlerle 2009 Konferansı Tutanakları
Hesaplamalı Dilbilim için İşleme Derneği, s 813-821
Snyder B, Palmer M (2004) İngilizce tüm kelimeleri görev. In: ACL 2004 Senseval-3 Work-
shop, Barcelona, Spain, URL http://www.cse.unt.edu/rada/senseval/senseval3/
yargılamanın / pdf / snyder.pdf
Steedman M (2000) Sintaktik Süreci. MIT Press, Cambridge, MA