Author manuscript, published in "The 23rd International Conference on Computational Linguistics - COLING 2010 (2010)"

inria-00536022, version 1 - 15 Nov 2010

Benchmarking for syntax-based sentential inference

Paul Bedaride
INRIA/LORIA
Université Henri Poincaré
paul.bedaride@loria.fr

Claire Gardent
CNRS/LORIA
claire.gardent@loria.fr

Abstract

We propose a methodology for investigating how well NLP systems handle meaning preserving syntactic variations. We start by presenting a method for the semi automated creation of a benchmark where entailment is mediated solely by meaning preserving syntactic variations. We then use this benchmark to compare a semantic role labeller and two grammar based RTE systems. We argue that the proposed methodology (i) supports a modular evaluation of the ability of NLP systems to handle the syntax/semantic interface and (ii) permits focused error mining and error analysis.

1 Introduction

First launched in 2005, the Recognising Textual Inference Challenge (RTE)[1] aims to assess in how far computer systems can emulate a human being in determining whether a short text fragment H referred to as the hypothesis, follows from or is contradicted by a text fragment T. In the RTE benchmarks, the hypothesis is a short constructed sentence whilst the text fragments are short passages of naturally occurring texts. As a result, the RTE challenge permits evaluating the capacity of NLP systems to handle local textual inference on real data, an enabling technology for any applications involving document interpretation.

In this paper, we focus on entailments based on meaning entailing, syntactic transformations such as:

(1) The man gives the woman the flowers that smell nice ⇒ The flowers which are given to the woman smell nice

[1] http://www.pascal-network.org/Challenges/RTE

We start (Section 2) by motivating the approach. We argue that the proposed evaluation methodology (i) interestingly complements the RTE challenge in that it permits a modular, analytic evaluation of the ability of NLP systems to handle syntax-based, sentential inference and (ii) permits focused error mining and analysis .

In Section 3, we go on to describe the benchmark construction process. Each item of the constructed benchmark associates two sentences with a truth value (true or false) indicating whether or not the second sentence can be understood to follow from the first. The construction of these benchmark items relies on the use of a grammar based surface realiser and we show how this permits automatically associating with each inference item, an entailment value (true or false) and a detailed syntactic annotation reflecting the syntactic constructs present in the two sentences constituting each benchmark item.

In section 4, we use the benchmark to evaluate and compare three systems designed to recognise meaning preserving syntactic variations namely, a semantic role labeller, Johan Bos' Nutcracker RTE system (where the syntax/semantic interface is handled by a semantic construction module working on the output of combinatory categorial grammar parser) and the Afazio system, a hybrid system combining statistical parsing, symbolic semantic role labelling and sentential entailment detection using first order logic. We give the evaluation figures for each system. Additionally, we show how the detailed syntactic annotations automatically associated with each benchmark item by the surface realiser can be used to identify the most likely source of errors that is, the syntactic constructs that most frequently co-occur with en entailment recognition error.

inria-00536022, version 1 - 15 Nov 2010

2 Motivations

Arguably focusing on meaning entailing syntactic transformations is very weak. Indeed, one of the key conclusions at the second RTE Challenge Workshop was that entailment modeling requires vast knowledge resources that correspond to different types of entailment reasoning e.g., ontological and lexical relationships, paraphrases and entailment rules, meaning entailing syntactic transformations and last but not least, world knowledge. Further, Manning (2006) has strongly argued against circumscribing the RTE data to certain forms of inference such as for instance, inferences based solely on linguistic knowledge. Finally, it is also often insisted that naturally occurring data should be favored over constructed data.

While we agree that challenges such as the RTE challenge are useful in testing systems abilities to cope with real data, we believe there is also room for more focused evaluation setups.

Focusing on syntax based entailments. As mentioned above, syntax based entailment is only one of the many inference types involved in determining textual entailment. Nevertheless, a manual analysis of the RTE1 data by (Vanderwende et al., 2005) indicates that 37% of the examples could be handled by considering syntax alone. Similarly, (Garoufi, 2007) shows that 37.5% of the RTE2 data does not involve deep reasoning and more specifically, that 33.8% of the RTE2 data involves syntactic or lexical knowledge only. Hence although the holistic, blackbox type of evaluation practiced in the RTE challenge is undeniably useful in assessing the ability of existing systems to handle local textual inference, a more analytic, modular kind of evaluation targeting syntax-based entailment reasoning is arguably also of interest.

Another interesting feature of the SSI (syntax-based sentential entailment) task we propose is that it provides an alternative way of evaluating semantic role labelling (SRL) systems. Typically, the evaluation of SRL systems relies on a hand annotated corpus such as PropBank or the FrameNet corpus. The systems precision and recall are then computed w.r.t. this reference corpus. As has been repeatedly argued (Moll and Hutchinson, 2003; Galliers and Jones, 1993), intrinsic evaluations may be of very limited value. For semantically oriented tools such as SRL systems, it is important to also assess their results w.r.t. the task which they are meant support namely reasoning : Do the semantic representations built by SRL help in making the correct inferences ? Can they be used, for instance, to determine whether a given sentence answers a given question ? or whether the content of one sentence follow from that another ? As explained in (Giampiccolo et al., 2007), entailment recognition is a first, major step towards answering these questions. Accordingly, instead of comparing the representations produced by SRL systems against a gold standard, the evaluation scheme presented here, permits evaluating them w.r.t. their ability to capture syntax based sentential inference.

It is worth adding that, although the present paper focuses on entailments strictly based on syntax, the proposed methodology should straightforwardly extend to further types of entailment such as in particular, entailments involving lexical relations (synonymy, antonymy, etc.) or entailments involving more complex semantic phenomena such as the interplay between different classes of complement taking verbs, polarity and author commitment discussed in (Nairn et al., 2006). This is because as we shall see in section 3, our approach is based on an extensive, hand written grammar of English integrating syntax and semantics. By modifying the grammar, the lexicon and/or the semantics, data of varying linguistic type and complexity can be produced and used for evaluation.

Hand constructed vs. naturally occurring data. Although in the 90s, hand tailored testsuites such as (Lehmann et al., 1996; Cooper et al., 1995) were deemed useful for evaluating NLP systems, it is today generally assumed that, for evaluation purposes, naturally occurring data is best. We argue that constructed data can interestingly complement naturally occurring data.

To start with, we agree with (Crouch et al., 2006; Cohen et al., 2008) that science generally benefits from combining laboratory and field studies and more specifically, that computational linguistics can benefit from evaluating systems on

inria-00536022, version 1 - 15 Nov 2010

a combination of naturally occurring and constructed data.

Moreover, constructed data need not be hand constructed. Interestingly, automating the production of this data can help provide better data annotation as well as better and better balanced data coverage than both hand constructed data and naturally occurring data. Indeed, as we shall show in section 4, the benchmark creation process presented here supports a detailed and fully automated annotation of the syntactic properties associated with each benchmark item. As shown in section 5, this in turn allows for detailed error mining making it possible to identify the most likely causes of system errors. Additionally, the proposed methodology permits controlling over such benchmark parameters as the size of the data set, the balance between true and false entailments, the correlation between word overlap and entailment value and/or the specific syntactic phenomena involved. This is in contrast with the RTE data collection process where "the distribution of examples is arbitrary to a large extent, being determined by manual selection[2]" (Giampiccolo et al., 2007). As has been repeatedly pointed out (Burchardt et al., 2007; Garoufi, 2007), the RTE datasets are poorly balanced w.r.t., both the frequency and the coverage of the various phenomena interacting with textual inference.

3 Benchmark

We now present the content of an SSI benchmark and the method for constructing it.

An SSI benchmark item (cf. e.g., Figure 1) consists of two sentences and a truth value (true or false) indicating whether or not the second sentence can be understood to follow from the first. In addition, each sentence is associated with a detailed syntactic annotation describing the syntactic constructs present in the sentence.

The benchmark construction process consists of two main steps. First, a generation bank is built. Second, this generation bank is drawn upon

T: The man gives the woman the flowers that smell nice
smell:{*n0Va1,active,relSubj,canAdj*}
give:{*n0Vn2n1,active,canSubj,canObj,canIObj*}
H: The flowers are given to the woman
give:{*n0Vn1Pn2,shortPassive,canSubj,canIObj*}
Entailment: TRUE

Figure 1: An SSI Benchmark item

to construct a balanced data set for SSI evaluation. We now describe each of these processes in turn.

Constructing a generation bank We use the term "generation bank" to refer to a dataset whose items are produced by a surface realiser i.e., a sentence generator. A surface realiser in turn is a program which associates with a given semantic representation, the set of sentences verbalising the meaning encoded by that representation. To construct our generation bank, we use the GenI surface realiser (Gardent and Kow, 2007). This realiser uses a Feature based Tree Adjoining Grammar (FTAG) augmented with a unification sematics as proposed in (Gardent and Kallmeyer, 2003) to produce all the sentences associated by the grammar with a given semantic representation. Interestingly, the FTAG used has been compiled out of a factorised representation and as a result, each elementary grammar unit (i.e., elementary FTAG tree) and further each parse tree, is associated with a list of items indicating the syntactic construct(s) captured by that unit/tree[3]. In short, GenI permits associating with a given semantics, a set of sentences and further for each of these sentences, a set of items indicating the syntactic construct(s) present in the syntactic tree of that sentence. For instance, the sentences and the syntactic constructs associated by GenI with the semantics given in (2) are those given in (3).

(2) A:give(B C D E) G:the(C) F:man(C) H:the(D) I:woman(D) J:the(E) K:flower(E) L:passive(B) L:smell(M E N) O:nice(N)

(3) a. The flower which smells nice is given to the woman by the man

[2]The short texts of the RTE benchmarks are automatically extracted from real texts using different applications (e.g., Q/A, summarisation, information extraction, information retrieval systems) but the query used to retrieve these texts is either constructed manually or post-edited.

[3]Space is lacking to give a detailed explanation of this process here. We refer the reader to (Gardent and Kow, 2007) for more details on how GenI associates with a given semantics, a set of sentences and for each sentence a set of items indicating the syntactic construct(s) present in the syntactic tree of that sentence.

inria-00536022, version 1 - 15 Nov 2010

give:n0Vn1Pn2-Passive-CanSubj-ToObj-ByAgt, smell:n0V-active-OvertSubjectRelative

b. The flower which smells nice is given the woman by the man
give:n0Vn2n1-Passive, smell:n0V-active-OvertSubjectRelative

c. The flower which is given the woman by the man smells nice
give:n0Vn2n1-Passive-CovertSubjectRelative, smell:n0V-active

d. The flower which is given to the woman by the man smells nice
give:n0Vn1Pn2-Passive-OvertSubjectRelative, smell:n0V-active

e. The flower that smells nice is given to the woman by the man
give:n0Vn1Pn2-Passive, smell:n0V-CovertSubjectRelative

f. The flower that smells nice is given the woman by the man
give:n0Vn2n1-Passive, smell:n0V-CovertSubjectRelative

g. The flower that is given the woman by the man smells nice
give:n0Vn2n1-Passive-CovertSubjectRelative, smell:n0V-active

h. The flower that is given to the woman by the man smells nice
give:n0Vn1Pn2-Passive-CovertSubjectRelative, smell:n0V-active

The tagset of syntactic annotation covers the subcategorisation type of the verb, a specification of the verb mood and a description of how arguments are realised.

The semantic representation language used is a simplified version of the flat semantics used in e.g., (Copestake et al., 2005) which is sufficient for the cases handled in the present paper. The grammar and therefore the generator, can however easily be modified to integrate the more sophisticated version proposed in (Gardent and Kallmeyer, 2003) and thereby provide an adequate treatment of scope.

Constructing an SSI benchmark. Given a generation bank, false and true sentential entailment pairs can be automatically produced by taking pairs of sentences $\langle S_1, S_2 \rangle$ and comparing their semantics: if the semantics of S_2 is entailed by the semantics of S_1, the pair is marked as TRUE else as FALSE. The syntactic annotations associated in the generation bank with each sentence are carried over to the SSI benchmark thereby ensuring that the overall information contained in each SSI benchmark is as illustrated in Figure 1 namely, two pairs of syntactically annotated sentences and a truth value indicating (non) entailment.

To determine whether a sentence textually entails another we translate their flat semantic representation into first order logic and check for logical entailment. Differences in semantic representations which are linked to functional surface differences such as active/passive or the presence/absence of a complementizer (*John sees Mary leaving/John sees that Mary leaves*) are dealt with by (automatically) removing the corresponding semantic literals from the semantic representation before translating it to first order logic. In other words, active/passive variants of the same sentence are deemed semantically equivalent.

Note that contrary to what is assumed in the RTE challenge, entailment is here logical rather than textual (i.e., determined by a human) entailment. By using logical, rather than textual (i.e., human based) entailment, it is possible that some cases of syntax mediated textual entailments are not taken into account. However, intuitively, it seems reasonable to assume that for most of the entailments mediated by syntax alone, logical and textual entailments coincide.

3.1 The SSI benchmark

Using the methodology just described, we first produced a generation bank of 226 items using 81 input formula distributed over 4 verb types. From this generation bank, a total of 6 396 SSI-pairs were built with a ratio of 42.6% true and 57.4% false entailments.

For our experiment, we extracted from this SSI-suite, 1000 pairs with an equal proportion of true and false entailments and a 7/23/30/40 distribution of four subcategorisation types namely, adjectival predicative (n0Va1 e.g., *The cake tastes good*), intransitive (n0V), transitive (n0Vn1) and ditransitive (n0Vn2n1)[4]. We furthermore con-

[4]The subcategorisation type of an SSI item is determined manually and refers either to the main verb if the sentence is

inria-00536022, version 1 - 15 Nov 2010

strained the suite to respect a neutral correlation between word overlap and entailment. Following (Garoufi, 2007), we define this correlation as follows. The word overlap $wo(T, H)$ between two sentences T and H is the ratio of common lemmas between T and H on the number of lemmas in H (non content words are ignored). If entailment holds, the word overlap/entailment correlation value of the sentence pair is $wo(T, H)$. Otherwise it is $1 - wo(T, H)$. The 1000 items of the SSI suite used in our experiment were chosen in such a way that the word overlap/entailment correlation value of the SSI suite is 0.49.

In sum, the SSI suite used for testing exhibits the following features. First, it is balanced w.r.t. entailment. Second, it displays good syntactic variability based both on the constrained distribution of the four subcategorisation types and on the use of the XTAG grammar to construct sentences from abstract representations (cf. the paraphrases in (3) generated by GenI from the representation given in (2)). Third, it contains 1000 items and could easily be extended to cover more and more varied data. Fourth, it is specifically tailored to check systems on their ability to deal with syntax based sentential entailment: word overlap is high, syntactic variability is provided and the correlation between word overlap and entailment is not biased.

4 System evaluation and comparison

SRL and grammar based systems equipped with a compositional semantics are primary targets for an SSI evaluation. Indeed these systems aim to abstract away from syntactic differences by producing semantic representations of a text which capture predicate/argument relations independent of their syntactic realisation.

We evaluated three such systems on the SSI benchmark namely, NutCracker, (Johansson and Nugues, 2008)'s Semantic Role Labeller and the Afazio RTE system.

4.1 Systems

Nutcracker Nutcracker is a system for recognising textual entailment which uses deep semantic processing and automated reasoning. Deep semantic processing associates each sentence with a Discourse Representation Structure (DRS (Kamp and Reyle, 1993)) by first, using a statistical parser to build the syntactic parse of the sentence and second, using a symbolic semantic construction module to associate a DRS with the syntactic parse. Entailment between two DRSs is then checked by translating this DRS into a first-order logical (FOL) formula and first trying to find a proof. If a proof is found then the entailment is set to true. Otherwise, Nutcracker backs off with a word overlap module computed over an abstract representation of the input sentences and taking into account WordNet related information. Nutcracker was entered in the first RTE challenge and scored an accuraccy (percentage of correct judgments) of 0.562 when used as is and 0.612 when combined with machine learning techniques. For our experiment, we use the online version of Nutcracker and the given default parameters.

Afazio Like Nutcracker, the Afazio system combines a statistical parser (the Stanford parser) with a symbolic semantic component. This component pipelines several rewrite modules which translate the parser output into a first order logic formula intended to abstract away from surface differences and assign syntactic paraphrases the same representation (Bedaride and Gardent, 2009). Special emphasis is placed on capturing syntax based equivalences such as syntactic (e.g., active/passive) variations, redistributions and noun/verb variants. Once the parser output has been normalised into predicate/argument representations capturing these equivalences, the resulting structures are rewritten into first order logic formulae. Like Nutcracker, Afazio checks entailment using first order automated reasoners namely, *Equinox* and *Paradox* [5].

SRL (Johansson and Nugues, 2008)'s semantic role labeller achieved the top score in the closed CoNLL 2008 challenge reaching a labeled semantic F1 of 81.65. To allow for comparison with Nutcracker and Afazio, we adapted the

a clause or to the embedded verb if the sentence is a complex sentence.

[5] http://www.cs.chalmers.se/~koen/folkung/

inria-00536022, version 1 - 15 Nov 2010

rewrite module used in Afazio to rewrite Predicate/Argument structures into FOL formula in such a way as to fit (Johansson and Nugues, 2008)'s SRL output. We then use FOL automated reasoner to check entailment.

4.2 Evaluation scheme and results

The results obtained by the three systems are summarised in Table 1. TP (true positives) is the number of entailments recognised as such by the system and TN (true negatives) of non entailments. Conversely, FN and FP indicate how often the systems get it wrong: FP is the number of non entailments labelled as entailments by the system and FN, the number of entailments labelled as non entailments. 'ERROR' refers to cases where the CCG parser used by Nutcracker fails to find a parse. The last three columns indicate the overall ability of the systems to recognise false entailments (TN/N with N the number of false entailment in the benchmark), true entailments (TP/P) and all true and false entailment (Precision).

Overall, Afazio outperforms both Nutcracker and the SRL system. This is unsurprising since contrary to these other two systems, Afazio was specifically designed to handle syntax based sentential entailment. Its strength is that it combines a full SRL system with a semantic construction module designed for entailment detection. More surprisingly, the CCG parser used by Nutcracker often fails to find a parse.

The SRL system has a high rate of false negatives. Using the error mining technique presented in the next section, we found that the most suspicious syntactic constructs all included a relativised argument. A closer look at the analyses showed that this was due to the fact that SRL systems fail to identify the antecedent of a relative pronoun, an identification that is necessary for entailment checking. Another important difference with Afazio is that the SRL system produces a single output. In contrast, Afazio checks entailment for any of the pairs of semantic representations derived from the first 9 parses of the Stanford parser. The number 9 was determined empirically and proved to yield the best results overall although as we shall see in the error mining section, taking such a high number of parses into account often leads to incorrect results when the hypothesis (H) is short.

Nutcracker, on the other hand, produces many false positives. This is in part due to cases where the time bound is reached and the word overlap backoff triggered. Since the overall word overlap of the SSI suite is high, the backoff often predicts an entailment where in fact there is none (for instance, the pair *'John gave flowers to Mary/Mary gave flowers to John* has a perfect word overlap but entailment does not hold). When removing the backoff results i.e., when assigning all backoff cases a negative entailment value, overall precision approximates 60%. In other words, on cases such as those present in the SSI benchmark where word overlap is generally high but the correlation between word overlap and entailment value is neutral, Nutcracker should be used without backoff.

5 Finding the source of errors

The annotations contained in the automatically constructed testsuite can help identify the most likely sources of failures. We use (Sagot and de La Clergerie, 2006)'s suspicion rate to compute the probability that a given pair of sets of syntactic tags is responsible for an RTE detection failure. The tag set pairs with highest suspicion rate indicate which syntactic phenomena often cooccurs with failure.

More specifically, we store for each testsuite item (T,H), all tag pairs (t_j, h_k) such that the syntactic tags t_j and h_k are associated with the same predicate P_i but t_j occurs in T and h_k in H. That is, we collect the tag pairs formed by taking the tags that label the occurrence of the same predicate on both sides of the implication. If a predicate occurs only in H then for each syntactic tag h_k labelling this predicate, the pair (nil, h_k) is created. Conversely, if a predicate occurs only in T, the pair (t_j, nil) is added. Furthermore, the tags describing the subcategorisation type and the form of the verb are grouped into a single tag so as to reduce the tagset and limit data sparseness. For instance, given the pair of sentences in Figure (1), the following tag pairs are produced:
(n0Va1:active:relSubj, nil)
(n0Va1:active:canAdj, nil)

inria-00536022, version 1 - 15 Nov 2010

system	ERROR	TN	FN	TP	FP	TN/N	TP/P	Prec
afazio	0	360	147	353	140	0.7200	0.7060	71.3%
nutcracker	155	22	62	312	449	0.0467	0.8342	39.5% (60% w/o B.O.)
srl	0	487	437	63	13	0.9740	0.1260	55.0%

Table 1: Results of the three systems on the SSI-testsuite (TN = true negatives, FN = false negatives, TP = true positives, FP = false positives, N = TN + FP, P = TP + FN, Prec = Precision, ERROR: no parse tree found)

(n0Vn2n1:active:canSubj,n0Vn1Pn2:shortPassive:canSubj)
(n0Vn2n1:active:canSubj,n0Vn1Pn2:shortPassive:canIObj)
(n0Vn2n1:active:canObj,n0Vn1Pn2:shortPassive:canSubj)
(n0Vn2n1:active:canObj,n0Vn1Pn2:shortPassive:canIObj)
(n0Vn2n1:active:canIObj,n0Vn1Pn2:shortPassive:canSubj)
(n0Vn2n1:active:canIObj,n0Vn1Pn2:shortPassive:canIObj)

For each tag pair, we then compute the suspicion rate of that pair using (Sagot and de La Clergerie, 2006)'s fix point algorithm. To also take into account pairs of sets of tags (rather than just pairs of single tags), we furthermore preprocess the data according to (de Kok et al., 2009)'s proposal for handling n-grams.

Computing the suspicion rate of a tag pair. The error mining's suspicion rate algorithm of (Sagot and de La Clergerie, 2006) is a fix point algorithm used to detect the possible cause of parsing failures. We apply this algorithm to the pair of annotated sentences resulting from running the three systems on the automatically created testsuite as follows. Each such pair consists of a pair of sentences, a set of tag pairs, an entailment value (true or false) and a result value namely FP (false positive), FN (false negative), TP (true positive) or TN (true negative). To search for the most likely causes of failure, we consider separately entailments from non entailments. If entailment holds, the suspicion rate of a sentence pair is 0 for true positive and 1 for false positives. Conversely, if entailment does not hold, the suspicion rate of the sentence pair is 0 for true negatives and 1 for false negatives.

The aim is to detect the tag pair most likely to make entailment detection fail[6]. The algorithm iterates between tag pair occurrences and tag pair forms, redistributing probabilities with each iteration as follows. Initially, all tag pair occurrences in a given sentence have the same suspicion rate namely, the suspicion rate of the sentence (1 if the entailment could not be recognised, 0 otherwise) divided by the number of tag pair occurrences in that sentence. Next, the suspicion rate of a tag pair form is defined as the average suspicion rate of all occurrences of that tag pair. The suspicion rate of a tag pair occurrence within each particular sentence is then recalculated as the suspicion rate of that tag pair form normalised by the suspicion rates of the other tag pair forms occurring within the same sentence. The iteration stops when the process reaches a fixed point where the suspicion rates have stabilised.

Extending the approach to pairs of tag sets. To account for entailment recognition due to more than one tag pair, we follow (de Kok et al., 2009) and introduce a preprocessing step which first, expands tag pair unigrams to tag pair n-grams when there is evidence that it is useful that is, when an n-gram has a higher suspicion rate than each of its sub n-grams. For this preprocessing, the suspicion of a tag pair t is defined as the ratio of t occurrences in unrecognised entailments and the total number of t occurrences in the corpus. To compensate for data sparseness, an additional expansion factor is used which depends on the frequency of an n-gram and approaches one for higher frequency. In this way, long n-grams that have low frequency are not favoured. The longer the n-gram is, the more frequent or the more suspicious it needs to be in order to be selected by the preprocessing step.

We apply this extension to the SSI setting. We first extend the set of available tag pairs with tag set pairs such that the suspicion rate of these pairs is higher that the suspicion rate of each of the smaller tagset pairs that can be constructed from these sets. We then apply (Sagot and de La Clerg-

[6]We make the simplifying hypothesis that for each entailment not recognised, a single tag pair or tag pair n-gram is the cause of the failure.

inria-00536022, version 1 - 15 Nov 2010

n0Vs1:act:CanSubj	nil	0.85
n0Vn1:act:CanObj	nil	0.46
n0V:betaVn	nil	0.28

Table 2: The first 3 suspects for false positives

n0V:act	n0V:act:RelCSubj	0.73
n0Vs1:act:CanSubj n0V:act:RelOSubj	n0Vs1:act:CanSubj n0V:betaVn	0.69
n0Vs1:act:CanSub n0V:act:CanSubj	n0Vs1:act:CanSubj n0V:betaVn	0.69

Table 3: The first 3 suspects for false negatives

erie, 2006)'s fix point algorithm to compute the suspicion rate of the resulting tag pairs and tag sets pairs.

Results and discussion. We now show how error mining can help shed some light on the most probable sources of error when using Afazio.

For false positives (non entailment labelled as entailment by Afazio), the 3 most suspect tag pairs are given in Table 2. The first pair (n0Vs1:act:CanSubj,nil) points out to cases such as *Bill sees the woman give the flower to the man / The man gives the flower to the woman.* where T contains a verb with a sentential argument not present in H. In such cases, we found that the sentential argument in T is usually incorrectly analysed, the analyses produced are fragmented and entailment goes through. Similarly, the second suspect (n0Vn1:act:CanObj,nil) points to cases such as *a man sees Lisa dancing / a man dances*, where the transitive verb in T has no counterpart in H. Here the high number of analyses relied on by Afazio together with the small size of H leads to entailment detection: because we consider many possible analyses for T and H and because H is very short, one pair of analyses is found to match. Finally, the third suspect (n0V:betaVn,nil) points to cases such as *Bill insists for the singing man to dance / Bill dances* where the gerund is wrongly analysed and a relation is incorrectly established by the parser between *Bill* and *dance* (in H).

For false negatives, the first suspect indicates incorrect analyses for cases where an intransitive with canonical subject in H is matched by an intransitive with covert relative subject (e.g., *Bill sees the woman give the flower to the man / the man gives the flower to the woman.*). The second suspect points to cases such as *Bill insists for the man who sings to dance / Bill insists that the singing man dances.* where an embedded verb with relative overt subject in T (sings) is matched in H by an embedded gerund. Similarly, the third suspect points to embedded verbs with canonical subject matched by gerund verbs as in *the man who Bill insists that dances sings / Bill insists that the singing man dances*.

6 Conclusion

The development of a linguistically principled treatment of the RTE task requires a clear understanding of the strength and weaknesses of RTE systems w.r.t. to the various types of reasoning involved. The main contribution of this paper is the specification of an evaluation methodology which permits a focused evaluation of syntax based reasoning on arbitrarily many inputs. As the results show, there is room for improvment even on that most basic level. In future work, we plan to extend the approach to other types of inferences required for textual entailment recognition. A more sophisticated compositional semantics in the grammar used by the sentence generator would allow for entailments involving more complex semantic phenomena such as the interplay between implicative verbs, polarity and downward/upward monotonicity discussed in (Nairn et al., 2006). For instance, it would allow for sentence pairs such as *Ed did not forget to force Dave to leave / Dave left* to be assigned the correct entailment value.

References

Bedaride, P. and C. Gardent. 2009. Noun/verb entailment. In *4th Language and Technology Conference*, Poznan, Poland.

Burchardt, A., N. Reiter, S. Thater, and A. Frank. 2007. A semantic approach to textual entailment: System evaluation and task analysis. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 10–16.

Cohen, K., W. Baumgartner, and L. Hunter. 2008. Software testing and the naturally occurring data assumption in natural language processing. In *Proc. of "Software engineering, testing, and quality assurance for natural language processing ACL Workshop"*.

inria-00536022, version 1 - 15 Nov 2010

Cooper, R., R. Crouch, J. van Eijck, C. Fox, J. van Genabith, J. Jaspars, H. Kamp, M. Pinkal, D. Milward, M. Poesio, and S. Pulman. 1995. A framework for computational semantics, FraCaS. Technical report. MS. Stanford University.

Copestake, A., D. Flickinger, C. Pollard, and I. A. Sag. 2005. Minimal recursion semantics: an introduction. *Research on Language and Computation*, 3.4:281–332.

Crouch, R., L. Karttunen, and A. Zaenen. 2006. Circumscribing is not excluding: A reply to manning. MS. Palo Alto Research Center.

de Kok, D., J. Ma, and G. van Noord. 2009. A generalized method for iterative error mining in parsing results. In *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks (GEAF 2009)*, pages 71–79, Suntec, Singapore, August. Association for Computational Linguistics.

Galliers, J. R. and K. Sparck Jones. 1993. Evaluating natural language processing systems. Technical report, Computer Laboratory, University of Cambridge. Technical Report 291.

Gardent, C. and L. Kallmeyer. 2003. Semantic construction in ftag. In *Proceedings of the 10th meeting of the European Chapter of the Association for Computational Linguistics*, Budapest, Hungary.

Gardent, C. and E. Kow. 2007. A symbolic approach to near-deterministic surface realisation using tree adjoining grammar. In *ACL07*.

Garoufi, K. 2007. Towards a better understanding of applied textual entailment: Annotation and evaluation of the rte-2 dataset. Master's thesis, Saarland University, Saarbrcken.

Giampiccolo, D., B. Magnini, I. Dagan, and B. Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

Johansson, R. and P. Nugues. 2008. Dependency-based syntactic-semantic analysis with propbank and nombank. In *CoNLL '08: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 183–187, Morristown, NJ, USA. Association for Computational Linguistics.

Kamp, H. and U. Reyle. 1993. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer.

Lehmann, S., S. Oepen, H. Baur, O. Lbdkan, and D. Arnold. 1996. tsnlp — test suites for natural language processing. In *In J. Nerbonne (Ed.), Linguistic Databases*. CSLI Publications.

Manning, C. D. 2006. Local textual inference: It's hard to circumscribe, but you know it when you see it - and nlp needs it. MS. Stanford University.

Moll, D. and B. Hutchinson. 2003. Intrinsic versus extrinsic evaluations of parsing systems. In *Proceedings European Association for Computational Linguistics (EACL), workshop on Evaluation Initiatives in Natural Language Processing*, Budapest.

Nairn, R., C. Condoravdi, and L. Kartunen. 2006. Computing relative polarity for textual inference. In *Proceedings of ICoS-5 (Inference in Computational Semantics)*, Buxton, UK.

Sagot, B. and E. de La Clergerie. 2006. Error mining in parsing results. In *Proceedings of ACL-CoLing 06*, pages 329–336, Sydney, Australie.

Vanderwende, L., D. Coughlin, and B. Dolan. 2005. What syntax can contribute in entailment task. In *Proceedings of the First PASCAL RTE Workshop*, pages 13–17.