Believe It or Not: Solving the TAC 2009 Textual Entailment Tasks through an Artificial Believer System

Ralf Krestel
L3S Research Center
Universität Hannover
Germany
krestel@L3S.de

René Witte and **Sabine Bergler**
Department of Computer Science
and Software Engineering
Concordia University, Montréal, Canada
witte|bergler@cse.concordia.ca

Abstract

The Text Analysis Conference (TAC) 2009 competition featured a new textual entailment search task, which extends the 2008 textual entailment task. The goal is to find information in a set of documents that are entailed from a given statement. Rather than designing a system specifically for this task, we investigated the adaptation of an existing *artificial believer* system to solve this task. The results show that this is indeed possible, and furthermore allows to recast the existing, divergent tasks of textual entailment and automatic summarization under a common umbrella.

1 Introduction

Filtering out useful information from a sea of content is a daily challenge for almost every individual. Little progress has been made in recent years in providing end users with tools that go beyond information retrieval (Witte and Gitzinger, 2009). Summarization in its various forms (including focused summaries and update summaries) has the potential to reduce the workload of an individual by providing compressed text views. Textual entailment recognition is an enabling technology that can improve precision and recall for many NLP tasks, for example, by detecting redundant information in a summary on a semantic rather than a syntactic level.

The idea of an *artificial believer* is to acquire knowledge by processing information (e.g., texts) and maintain a consistent belief base by detecting inconsistencies and rejecting information according to a belief revision strategy. Our Fuzzy Believer application deploys a model of a human (newspaper) reader to decide which information to reject in case of inconsistencies; a long-term view of this approach is that of an artificial proxy that can process large amounts of content using a bias that reflects its human peer and allows access to its generated knowledge base by generating summaries or answering questions. To be able to build a consistent belief base, the Fuzzy Believer needs to be able to detect conflicting information. This subtask is isomorphic to recognizing textual entailment: to be detected as conflicting, the new information must (1) be recognized as pertaining to the same topic (2) not requiring a belief revision when adding it to the knowledge base. Then, summarizing a text can be seen as the task of generating a belief base by "reading" a set of documents, thereby removing redundant and conflicting information, and then generating a summary from this belief base. In the same form, the new TAC pilot RTE search task can be reinterpreted as the task of first believing (adding to the knowledge base) the hypothesis, then considering each sentence from the following documents. If a sentence belongs to the same topic and can be added without causing a revision, we include it in our search result.

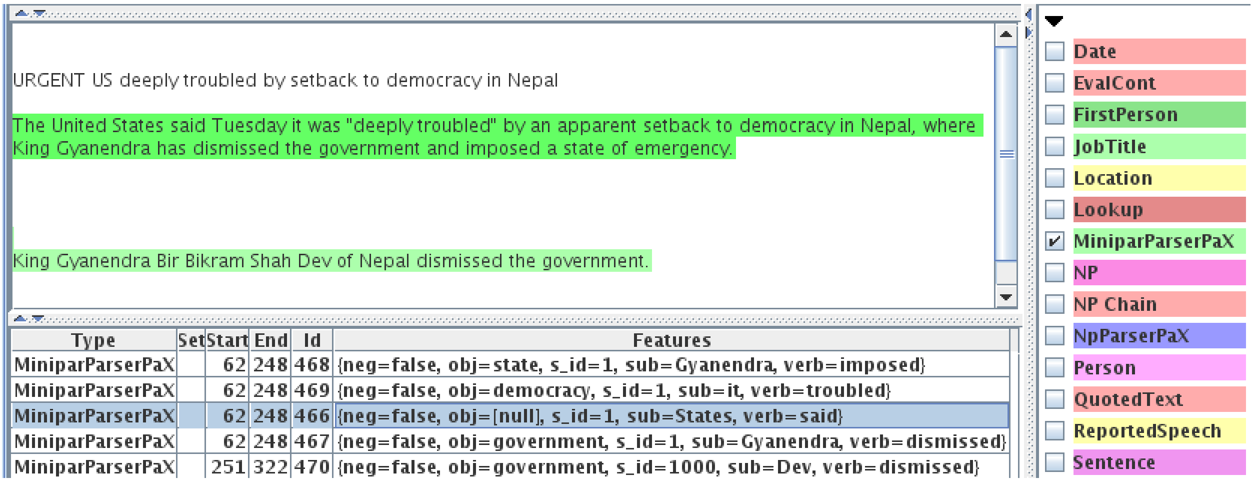
For the pilot task, we generated a document for each article together with the hypothesis in question. Figure 1 shows the result for one pair. We computed two results. One strict run required all three elements of the predicate-argument structures to match. One more lenient run required only two elements per PAS to match. In this example, we get a true positive result since one PAS from the sentence match with one PAS from the hypothesis for two elements (verb and object).

2 The RTE Pilot Task

The description of the pilot task[1] says:

> The *Textual Entailment Search Task* from TAC 2009 consists in finding all the sentences in a set of documents that entail a given hypothesis. The task is situated in the Summarization

[1] RTE-5 Search Pilot Guidelines, http://www.nist.gov/tac/2009/RTE/RTE5_Pilot_Guidelines.pdf



URGENT US deeply troubled by setback to democracy in Nepal

The United States said Tuesday it was "deeply troubled" by an apparent setback to democracy in Nepal, where King Gyanendra has dismissed the government and imposed a state of emergency.

King Gyanendra Bir Bikram Shah Dev of Nepal dismissed the government.

Type	Set	Start	End	Id	Features
MiniparParserPaX		62	248	468	{neg=false, obj=state, s_id=1, sub=Gyanendra, verb=imposed}
MiniparParserPaX		62	248	469	{neg=false, obj=democracy, s_id=1, sub=it, verb=troubled}
MiniparParserPaX		62	248	466	{neg=false, obj=[null], s_id=1, sub=States, verb=said}
MiniparParserPaX		62	248	467	{neg=false, obj=government, s_id=1, sub=Gyanendra, verb=dismissed}
MiniparParserPaX		251	322	470	{neg=false, obj=government, s_id=1000, sub=Dev, verb=dismissed}

- ☐ Date
- ☐ EvalCont
- ☐ FirstPerson
- ☐ JobTitle
- ☐ Location
- ☐ Lookup
- ☑ MiniparParserPaX
- ☐ NP
- ☐ NP Chain
- ☐ NpParserPaX
- ☐ Person
- ☐ QuotedText
- ☐ ReportedSpeech
- ☐ Sentence

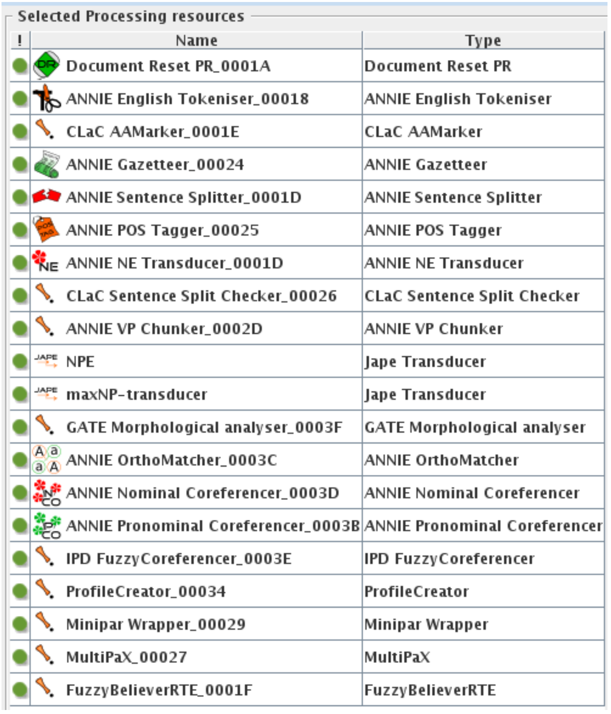
Figure 1: GATE screenshot of the results using MiniPar for one pair of sentences in the pilot task

application setting, where the hypothesis (H) is taken from a Summary Content Unit1 (SCU), and the systems must find all the entailing sentences (Ts) in a corpus of 10 newswire documents about a common topic.

The difference to previous textual entailment detection tasks and this year's main task is the additional context for each text (T) and hypothesis (H) pair. They are not isolated in an artificial way but rather embedded within a document. This includes possible references to entities, events, dates, places, situations, etc. pertaining to the topic.

3 The Fuzzy Believer Approach

In (Krestel et al., 2007a; Krestel et al., 2007b) we presented a system to model beliefs extracted from reported speech in newspaper articles. We deployed the system to detect textual entailment (Krestel et al., 2008) and inferred for the RTE challenge whether a statement "entails" or "contradicts" another statement or whether it is "unknown" in case we cannot make a reliable guess for one of the first two. The whole process is modeled in the context of fuzzy set theory. Each statement is represented by its predicate-argument structures (PAS), typically triples in the form of (subject, predicate, object). We use a couple of heuristics to compare these predicate-argument structures. If we obtain a similarity score higher than a threshold after merging two fuzzy predicate-argument structures, we conclude that the first statement entails the second. Similar for contradiction where we use some negative heuristics. If the PAS from the hypothesis do not match any of the PAS in the text, we label the relation as unknown. Figure 2 shows the GATE pipeline for generating the TAC-RTE results using MiniPar.



Selected Processing resources

!	Name	Type
●	Document Reset PR_0001A	Document Reset PR
●	ANNIE English Tokeniser_00018	ANNIE English Tokeniser
●	CLaC AAMarker_0001E	CLaC AAMarker
●	ANNIE Gazetteer_00024	ANNIE Gazetteer
●	ANNIE Sentence Splitter_0001D	ANNIE Sentence Splitter
●	ANNIE POS Tagger_00025	ANNIE POS Tagger
●	ANNIE NE Transducer_0001D	ANNIE NE Transducer
●	CLaC Sentence Split Checker_00026	CLaC Sentence Split Checker
●	ANNIE VP Chunker_0002D	ANNIE VP Chunker
●	NPE	Jape Transducer
●	maxNP-transducer	Jape Transducer
●	GATE Morphological analyser_0003F	GATE Morphological analyser
●	ANNIE OrthoMatcher_0003C	ANNIE OrthoMatcher
●	ANNIE Nominal Coreferencer_0003D	ANNIE Nominal Coreferencer
●	ANNIE Pronominal Coreferencer_0003B	ANNIE Pronominal Coreferencer
●	IPD FuzzyCoreferencer_0003E	IPD FuzzyCoreferencer
●	ProfileCreator_00034	ProfileCreator
●	Minipar Wrapper_00029	Minipar Wrapper
●	MultiPaX_00027	MultiPaX
●	FuzzyBelieverRTE_0001F	FuzzyBelieverRTE

Figure 2: GATE screenshot of the Fuzzy Believer pipeline

3.1 Updates for TAC 2009

For 2009 we improved our system a little bit. In addition to using the predicate-argument structures extracted from one of the parsers (SUPPLE, MiniPar, RASP, Stanford Parser) using PAX[2] we also made use of a noun

[2] Predicate-Argument Extractor (PAX), http://www.semanticsoftware.info/pax

phrase extractor (MuNPEx).[3] Each noun phrase that contains a modifier was converted into a predicate-argument structure. For example, the phrase "the expensive fuel" becomes the PAS "fuel – is – expensive". This improves finding matches in the text for rather simple hypotheses like "Bobby Fischer is a chess master."

For the pilot task, we took the hypothesis and added all sentences in the text that contained predicate-argument structures where two/three elements were close enough to the elements in the PAS from the hypothesis. The distance between two elements was computed using two heuristics. One to match strings and one to measure the distance within WordNet.

In general, our approach is comparable with the three-way decision task when considering the hypothesis and each sentence in the documents as the text. For all cases where we have "unknown" or "contradiction" as an analysis result we remove the sentence from the candidate list and keep only the sentences where we have "entailment."

4 Evaluation

Evaluation was done using the standard measures accuracy, recall, precision, and F-measure. Besides the 2-way task where the systems have to decide whether a text entails the hypothesis or not, for the 3-way task the systems have to decide on entailment, contradiction or unknown. For the pilot task different averages where computed based on (1) each sentence T (micro), (2) the averages for each topic (macro topic), and (3) the averages for each hypothesis (macro hypo).

Accuracy	*Using MiniPar*			
	QA	*IE*	*IR*	*Overall*
2-way	0.54	0.53	0.61	0.56
3-way	0.48	0.45	0.53	0.49
Accuracy	*Using Stanford Parser*			
	QA	*IE*	*IR*	*Overall*
2-way	0.49	0.51	0.58	0.52
3-way	0.42	0.43	0.49	0.44

Table 1: Main Task: Accuracy of our system for 2 different parsers

Results from our system for the main task can be seen in Table 1. Using MiniPar yields for both, 2- and 3-way task, better results than using the Stanford Parser. In Table 2, the results for the pilot task can be found. Regarding the two parsers, using predicate-argument structures extracted from MiniPar is also better for this task. Being more strict and only considering PAS to match when all three PAS elements match increases precision significantly. Unfortunately, but not surprisingly, recall drops dramatically. The detailed results for our third run using the Stanford parser and requiring three PAS elements to match are shown in Table 3.

Average	*Using MiniPar*		
	Precision	*Recall*	*F-Measure*
micro	0.12	0.37	0.18
macro topic	0.13	0.36	0.19
macro hypo	0.18	0.39	0.24
Average	*Using Stanford*		
	Precision	*Recall*	*F-Measure*
micro	0.11	0.27	0.15
macro topic	0.12	0.28	0.17
macro hypo	0.13	0.32	0.18

Table 2: Pilot Task: Results of our system for two different parsers and two PAS elements need to match

Average	*Using Stanford*		
	Precision	*Recall*	*F-Measure*
micro	0.19	0.06	0.10
macro topic	0.36	0.08	0.13
macro hypo	0.18	0.10	0.12

Table 3: Pilot Task: Results of our system for Stanford Parser three PAS elements have to match

4.1 Ablation Tests

As external resource we only use WordNet in one of our heuristics. The ablation tests show that for the three-way task the influence of WordNet is rather negligible: Accuracy of 0.480 vs. 0.487. For the two-way task there is no difference in accuracy: 0.560 for both. The effect is only observable when looking at the confusion matrices. For the three-way task the results are shown in Table 4 with and without making use of WordNet. When WordNet was used, the correct entailment increased. Unfortunately, the rate of correctly classified "unknown" cases decreased in about the same magnitude.

5 Conclusions

The pilot task is a more native application scenario for our Fuzzy Believer system. Still, not all entailments can be found relying only on the output of parsers. The inclusion of noun phrases to generate predicate-argument structures showed promising results and allows to capture some previously undetected entailment relations. In future work, we plan to perform experiments with end users to investigate the practical application of these concepts and the performance they require for productive use.

[3] Multi-lingual Noun Phrase Extractor (MuNPEx), http://www.semanticsoftware.info/munpex

Without WordNet		System Response			
		Entailment	Unknown	Contradiction	Total
Gold Standard	Entailment	119	173	8	300
	Unknown	41	169	0	210
	Contradiction	42	48	0	90
	Total	202	390	8	600

With WordNet		System Response			
		Entailment	Unknown	Contradiction	Total
Gold Standard	Entailment	142	156	2	300
	Unknown	60	150	0	210
	Contradiction	46	44	0	90
	Total	248	350	2	600

Table 4: Confusion Matrix for the three-way task using MiniPar and 2 PAS elements have to match

References

Ralf Krestel, René Witte, and Sabine Bergler. 2007a. Creating a Fuzzy Believer to Model Human Newspaper Readers. In Z. Kobti and D. Wu, editors, *Proc. of the 20th Canadian Conference on Artificial Intelligence (Canadian A.I. 2007)*, LNAI 4509, pages 489–501, Montréal, Québec, Canada, May 28–30. Springer.

Ralf Krestel, René Witte, and Sabine Bergler. 2007b. Processing of Beliefs extracted from Reported Speech in Newspaper Articles. In *Proc. of Recent Advances in Natural Language Processing (RANLP-2007)*, Borovets, Bulgaria, September 27–29.

Ralf Krestel, Sabine Bergler, and René Witte. 2008. A Belief Revision Approach to Textual Entailment Recognition. In *Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA, November 17-19. National Institute of Standards and Technology (NIST).

René Witte and Thomas Gitzinger. 2009. Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients. In *3rd Asian Semantic Web Conference (ASWC 2008)*, volume 5367 of *LNCS*, pages 360–374, Bangkok, Thailand, February 2–5. Springer.