A hierarchical taxonomy for classifying hardness of inference tasks

Martin Gleize, Brigitte Grau

A hierarchical taxonomy for classifying hardness of inference tasks

Martin Gleize, Brigitte Grau
LIMSI-CNRS
B.P. 133 91403 ORSAY CEDEX, France
gleize@limsi.fr, bg@limsi.fr

Abstract

Exhibiting inferential capabilities is one of the major goals of many modern Natural Language Processing systems. However, if attempts have been made to define what textual inferences are, few seek to classify inference phenomena by difficulty. In this paper we propose a hierarchical taxonomy for inferences, relatively to their hardness, and with corpus annotation and system design and evaluation in mind. Indeed, a fine-grained assessment of the difficulty of a task allows us to design more appropriate systems and to evaluate them only on what they are designed to handle. Each of seven classes is described and provided with examples from different tasks like question answering, textual entailment and coreference resolution. We then test the classes of our hierarchy on the specific task of question answering. Our annotation process of the testing data at the QA4MRE 2013 evaluation campaign reveals that it is possible to quantify the contrasts in types of difficulty on datasets of the same task.

Keywords: inference, question answering, textual entailment

1. Introduction

Exhibiting inferential capabilities is one of the major goals of many modern NLP systems dealing with question answering, information retrieval, information extraction or text summarization. In recent years, there has even been a focus on a new task called textual entailment, which precisely aims at capturing semantic inferences (Dagan et al., 2006).

However, inferences have been defined in several ways in the past, and it is not always easy to know the kind of problems we are faced with when dealing with textual entailment related tasks. In the context of machine reading, we can view an inference as any information we can reasonably deduce from the text without it being explicitly stated. This is different from logical inference and statistical inference, although techniques for both are used effectively in the resolution of textual inference problems (Moldovan et al., 2007; Poon, 2010).

This broad definition does not say much of what information we want to draw from the text, how to do it and whether it is an easy task or not. However, the ability to pinpoint the kind of inferential phenomena our systems are realistically able to handle seems really valuable to us. In this paper, we present a hierarchy of classes of inference. This hierarchical view is designed with several goals in mind: the most important is being able to categorize instances of problems to spot where the difficulties lie and what nature and extent of NLP techniques and resources we need to leverage to handle them. It also allows to conduct more fine-grained diagnostic evaluations of systems, to complement typical black-box evaluations.

We show in our experiments on QA4MRE 2013's multiple-choice questions that our hierarchical classes do capture the variations of type and hardness of inference problems for the task of question answering and provide a basis for experimenting and evaluating only on the data our systems are designed to handle.

2. Related works

To our knowledge, few attempts have been made to categorize the difficulty of inferences in a manner helpful to computational systems.

The field of cognitive psychology (McKoon and Ratcliff, 1992) distinguishes bridging and elaborative inferences. Bridging inferences are drawn to fill the gaps in text and explicit –often through access to world knowledge– what is untold, yet required to understand what the text means. This is akin to what the machine reading system of Peñas and Hovy (2010) does. Elaborative inferences are not required for textual coherence, but they serve to enrich our mental representation of the text and make it more memorable. They also are prominently used when answering questions on the text.

Such a distinction, although insightful to anticipate the design of machine reading and question answering systems, is still far from giving us hints on the automatization of inference. Clark et al. (2012) provide a combination of well-known language resources used toward the common goal of textual entailment. Ablation tests are performed to compute the impact of each on the overall accuracy, but there is no way of knowing which resource helps best on a given type of questions or entailment pairs.

Interestingly, Huang et al. (2013) take the angle of modeling human negative entailment capabilities. Finding in the text hints of a contradiction with the hypothesis appears to be an effective way to tackle textual entailment, but the tools and resources are still lacking to yield significant improvements over state of the art RTE systems.

MacCartney and Manning (2007) introduce natural logic applied to textual inference. They describe the kind of inference problems this logic applied to natural language is capable of solving: natural logic handles monotonicity, in which the concepts or constraints expressed are expanded or contracted, but it is not designed to deal with paraphrase, temporal reasoning, or relation extraction.

More generally, contributions which analyze the categories of error of systems are those we want to build upon.

Moldovan et al. (2003) measure their system's accuracy by question class: factual, simple-reasoning, fusion-list, interactive-context, speculative. However, these classes are more about question types than they are about their difficulty, even if the two are likely correlated.

3. The hierarchy

Each of our classes is built to encompass a range of natural language semantic problems, tools, techniques, resources, and even human cognitive processes and levels of world knowledge required. Practically, we say that a problem is of a given class if it can reliably be solved within that class, but not within the classes below it in the hierarchy. The aim is to capture the inference phenomena that are sufficient and necessary to solve the problem. In this respect, those classes can be used in the same way complexity classes are used: both to characterize problems and the systems solving them –a system is of a given class if it can solve problems of that class but not problems from higher classes.

While this all sounds ambitious, this hierarchy does not pretend to be absolute, and in fact it does not need to: simply setting up the groundwork towards a unified framework to discuss the contrasts in relative nature and difficulty inside the same problem –and sometimes the same dataset– is already helpful.

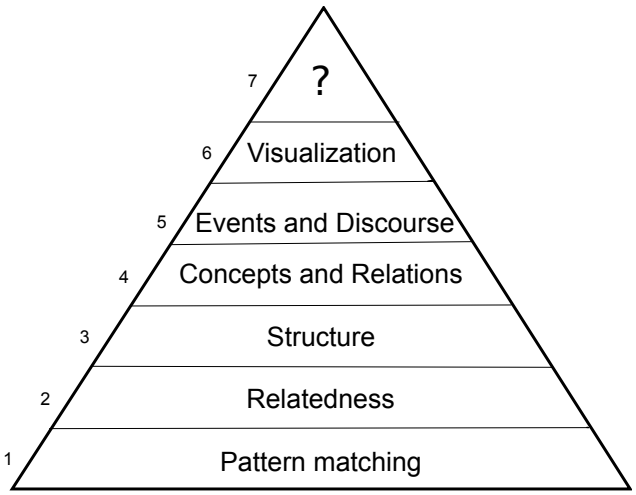

Figure 1: Hierarchy of inference classes

The hierarchy is shown in Figure 1. There are seven classes, and the first six (from bottom to top) pair up in three tiers, corresponding roughly to the units of sense considered. As members of the hierarchy, higher classes will often still rely on the techniques and resources featured in classes below, and it is reasonable to assume that a system of a given class behaves well when faced with problems of lower classes. We also expect the difficulty of the problems –and the error rate of the current state of the art methods– to generally increase with the class.

We propose for each class examples of representative problems. We chose to frame the problems, when possible, as picking the most relevant item among several answer candidates to a natural language query. While it seems reasonable to expect that our classes can still characterize the absolute difficulty level of the query alone, this level would be difficult to assess without some elements of context or examples of concurrent wrong items our systems are susceptible to pick over the right ones. Providing this context and an explicit set of correct and incorrect items is a satisfying compromise which allows us to showcase which techniques and resources are needed to discriminate reliably between possible answers to the query. In our examples, we will put the number of the right answer in bold font.

We accordingly specify how several well-known natural language tasks reduce to this common framework.

In question answering, the query can take the form of a natural language question, and items can be some choices of answer, the task being naturally to find a fitting answer to the question. Ranking answer candidates is also how state of the art question answering systems like IBM's Watson computer operate (Ferrucci et al., 2010), so this is a quite direct formulation of the task.

In coreference resolution, the query is a sentence containing a pronoun, associated with the text preceding it. The items are entities of the text and the task is to find which entity the pronoun refers to. Recently, Levesque et al. (2011) have argued that the problem of resolving the difficult pronouns in a carefully chosen set of sentences, which he refers to as the Winograd Schema Challenge, could serve as a conceptually and practically appealing alternative to the well-known Turing Test (Turing, 1950). According to Levesque, the pitfalls lie in the difficulty of providing problems whose resolution is obvious for humans but hard for machines. Knowing what is difficult for machines seems key to us and is one of the goals addressed in this paper. We will use examples of Winograd Schemas to demonstrate that coreference resolution can reach a very high difficulty level. A Winograd Schema is a small reading comprehension test involving the question of which of the two candidate antecedents for the definite pronoun in a given sentence is its correct antecedent. There is a word (called the *special* word) which appears in the text and sometimes the question. When it is replaced by another word (called the *alternate* word), the text still makes sense, but the answer to the question changes. In this special coreference task, we will set the query as the sentence and the question, and the candidate items as pairs of (special/alternate word, coreferent assignment), as shown for the classic Winograd scheme example in table 1.

Text: The city councilmen refused the demonstrators a permit because they [feared/advocated] violence.
Question: Who [feared/advocated] violence?
1) feared → councilmen, advocated → demonstrators
2) feared → demonstrators, advocated → councilmen

Table 1: Winograd schema sample

In student answer analysis, an applicative task to Recognizing Textual Entailment evaluated in SemEval 2013's Task 7 (Dzikovska et al., 2013), the query is a correct reference

answer to a question, and we choose as items correct and incorrect student answers. The task is to find correct student answers matching one reference answer. Some of our examples are sampled from Semeval 2013 Task 7's Beetle test data.
Other problems which may fit this framework are word sense disambiguation and entity linking.

3.1. Tier 1: Word level

3.1.1. Pattern matching

Class 1 pertains to problem instances that can be solved using only words of the text, without any intent to capture the representation of a higher-level structure like the sentence, and using minimal world knowledge. Sentence chunking to create smaller groups of words, filtering words of the text (with a stop-word list, for example), tokenization, grouping words in n-grams, basic stemming and lemmatization are the most advanced text processing needed to solve this class of problems.

Text: Bob is going to the swimming pool. Question: What is going to do Bob? 1) Eat a sandwich. **2)** Go swimming.

Table 2: Question Answering (Class 1)

Reference answer: Terminal 4 and the positive terminal are separated by the gap 1) Because they aren't damaged. **2)** positive battery terminal is separated by a gap from terminal 4

Table 3: Semeval 2013 Task 7 (Class 1)

These sample problems (tables 2 and 3) are easily solved through counting the number of words common to a choice and the given text. It is enough to discriminate between the right choice (choice 2) and something irrelevant.

3.1.2. Relatedness

Class 2 does not need the inference process to represent a higher level of sense than Class 1 (we stay at word level), and uses mostly the same processing tools, but we add the notion of lexico-semantic variations of words to capture a shallow notion of semantic relatedness. Lists of synonyms/hypernyms, taxonomies, thesauri –including WordNet (Miller, 1995) used in the simplest ways– and dictionaries are among common resources that are added in to deal with those problems.
In the example (table 4), counting the common words between the choices and the text is not enough and does not distinguish one answer from the other. But *hamburger* is an hyponym of *sandwich*. This can be learned from a good enough synonym/hypernym resource, like WordNet, or the Wiktionary [1].

Text: Bob bought this hamburger. Question: What has Bob done? **1)** Obtained a sandwich. 2) Played in the water.

Table 4: Question Answering (Class 2)

3.2. Tier 2: Sentence level

3.2.1. Structure

Class 3's inference problems require the capture of some notion of higher-level structure in the text, at least higher than the word, generally at sentence –or clause– level. Several of the most well-studied text processing techniques come in handy, like: part-of-speech tagging, chunking, syntactic parsing, predicate-argument extraction, semantic role labeling, or polarity detection. We note that we still often need techniques and resources mentioned in Tier 1, but we moreover need to know not only what the words are, but what roles they occupy relatively to the others in the sentence.

Text: As it's raining today, Bob won't go to the beach. Question: What is going to do Bob? **1)** Stay home. 2) Go to the beach with his friends.

Table 5: Question Answering (Class 3)

Reference answer: Terminal 4 and the positive terminal are separated by the gap **1)** terminal 4 is not connected to the positive battery terminal 2) because there is no gap between terminal four and the positive terminal

Table 6: Semeval 2013 Task 7 (Class 3)

The problem described in the examples is as expected harder than Tier 1 problems. Simple Tier 1 techniques can work against us and reliably pick the wrong choice. In table 5, we have to determine that no matter what, Bob will surely not *go to the beach* , by parsing the text and detecting that this clause is negated, so we pick the other answer. This is also an example of using negative entailment in the decision process. In example table 6, to pick the student answer corresponding to the reference answer (choice 1), we rely on polarity detection on some kind of predicate-argument structure extraction to establish typed links between the *terminal 4* and *the positive terminal*. Note that it is necessary to know that *separated* and *connected* are antonyms, which can be found in resources used in Class 2.

3.2.2. Concepts and relations

Similarly to the first two classes, Class 4 takes the complexity of Class 3 and adds in extended world knowledge. Establishing a link between a word –or sequence of words– and a real-world concept is now required, as well as using not only the relations between concepts discovered in

[1] `http://www.wiktionary.org/`

the text, but also those stored in a background knowledge base. Named entity recognition, paraphrases, ontologies, relations and properties extracted from dictionaries, WordNet or Wikipedia and even relations extracted through webcrawling (like TextRunner) are good techniques and resources to capture concepts and relations. Some basic reasoning engine may be needed.

Text: Bob once gave his sister Alice a pendant. Question: Who gave Alice the pendant? **1)** A brother. 2) A sister.

Table 7: Question Answering (Class 4)

Text: The lawyer asked the witness a question, but he was reluctant to [answer/repeat] it. Question: Who was reluctant to [answer/repeat] the question? **1)** answer → witness, repeat → lawyer 2) answer → lawyer, repeat → witness

Table 8: Winograd schema (Class 4)

In the question answering example (table 7), a system has to know that if a male individual has a sister, then he is the brother of that female individual, which is not linguistic knowledge easily obtainable in the previous classes.
In the Winograd scheme (table 8), there are two main ways to go about it. Either you know that the indirect object of the verb *ask* often has to answer the question after it is asked, which is quite complex reasoning that will be handled in class 5, or you just pick the character that is more likely to answer questions, when a *witness* and a *lawyer* are involved. For this method and this example, TextRunner gives as potential relations between *witness* and *question* only variants of *answer* or *ask* at the passive form, while relations found between *lawyer* and *question* are much more diverse, modeling the fact that the witness is often one who strictly answers questions.

3.3. Tier 3: Beyond the text

3.3.1. Events and discourse

Class 5's phenomena go beyond a single sentence and deal with characters and events in the text. In particular, understanding the overall structure of a sequence of several sentences is required: to know when an event might have happened without being mentioned in the text, what event might happen in the future, what roles are filling the characters. NLP techniques and resources at this stage are much more scarce. Discourse parsing can unveil simple causal or temporal relations between events. Event chains can help filling out the blanks in a succession of events. We feel like common-sense knowledge of human society and interactions would help at this stage, without needing the full extent of what Class 6 is about.

All of the choices in the example (table 9) are true statements, but are not the reason asked in the question. This question also deals with common states of mind of human beings.

Example from QA4MRE 2013, entrance exams task
Text: I probably would have [continued to argue with her], but as I lay there, I could tell that Susan's phone call was not good news. I knew she had a boyfriend back home. From what I could hear her say, I guessed he had found a new girlfriend. Question: Why was Susan so upset by the phone call? 1) Mary's boyfriend had found a new girlfriend. 2) The call interrupted her argument with Mary. **3)** Her boyfriend had lost interest in her. 4) The caller did not want to talk to Mary.

Table 9: Question Answering (Class 5)

Text: Susan knew that Ann's son had been in a car accident, [so/because] she told her about it. Who told the other about the accident? 1) so → Ann, because → Susan **2)** so → Susan, because → Ann

Table 10: Winograd schema (Class 5)

The Winograd schema of class 5 (table 10) requires us to keep track of 2 characters and their interaction in the text. Although technically all the information is included in the same sentence, there are multiple clauses to link together. Detecting a causal relationship –with reliable directionality– is a typical class 5 problem.

3.3.2. Visualization

Class 6 problems go beyond the text itself, and require an actual model of the situation at hand (Zwaan and Radvansky, 1998). A few dedicated NLP applications exist solely to solve one of the many facets of Class 6 inferences, including: temporal and spatial reasoning, sentiment detection. In general, computer systems are lacking human senses –vision and hearing– to deal with these problems in a generic fashion.

Example from QA4MRE 2013, main task
Text: 1 person in 85 will be affected [by Alzheimer's] by the year 2050. Question: How many people affected by Alzheimer's are there expected to be in the year 2050? **1)** 123 million. 2) 85 million. 3) none. 4) a pretty small number. 5) None of the above.

Table 11: Question Answering (Class 6)

The example from table 11 is one of the harder questions. One has to know or evaluate the predicted human population of Earth by 2050 and make a computation about that number. Several choices are provided so we can pick the likelier number, but the system still has to be aware of

Text: I tried to paint a picture of an orchard, with lemons in the lemon trees, but they came out looking more like [light bulbs / telephone poles]. Question: What looked like [light bulbs / telephone poles]? **1)** light bulbs → lemons, telephone poles → lemon trees 2) light bulbs → lemon trees, telephone poles → lemons

Table 12: Winograd schema (Class 6)

global growing demographics on Earth, and know how to perform simple calculations. Alternatively, one has to guess that the other likely number (85 million) is probably just a decoy because the number 85 is present in the text but not to designate a population count; this kind of meta-reasoning about the task format (multiple choice questions) is not currently handled by systems, and is not intended to be the main natural language problem to be addressed in the near future.

In the Winograd schema from table 12, it is much easier for a human to just imagine what fruits or trees might look like –relatively to their visual shape– than for a computer.

3.4. Leaving room for the unexpected

We leave room for more complex inference problems and techniques.

4. Experiments

Our data consist of the question answering test sets at QA4MRE 2013 (Peñas et al., 2013). There are two tasks, the Main task and the Entrance Exams task, and both feature the same format: a series of long texts, and for each of them, several multiple-choice questions to answer. The Main task's questions are traditionally designed to evaluate natural language processing systems. But the all new Entrance Exams task features tests of English as a foreign language at the Japanese University Entrance exams, hence this dataset is designed to evaluate humans. This is a key difference.

There are 284 questions over 4 topics of 4 reading documents each in the Main task, and 45 questions over 9 reading documents in the Entrance Exams task. Questions have 5 answer choices in the Main task (including a *None of the above* option to indicate that none of the provided answer choices is correct), 4 answer choices in the Exams task (a *None of the above* option is not present for those). Each question has been annotated with its correct answer (as provided by QA4MRE organizers) and our own annotations of a 3-sentence passage in the text containing a single answer sentence. We removed questions for which we couldn't find a correct passage: in particular, all questions where *None of the above* is the correct answer were filtered out (39% of the Main dataset).

Our goal is to annotate each question with their class in our hierarchy. We turn the question answering problem into 2 separate subtasks: first we need to find the passage containing the answer, and once this is done –essentially reducing the search space for the answer from hundreds of sentences to just three– we need to consider answer candidates present in this passage, and choose the correct one. Each of these tasks can be framed as described at the beginning of section 3.. For the passage retrieval subtask (PR), the query is the question and the candidate items are the passages of the text. For the answer choice subtask (AC), the query is the question and the 3-sentence passage –now assumed correct and containing the answer– and the candidate items are the answer choices. The overall difficulty of the question answering task (QA) can be approximated by taking the maximum of the difficulties of the two subtasks, passage retrieval and answer choice [2].

For convenience purposes, we run a baseline counting the common lemmatized non stop-words between candidate items and query and rank the candidate items according to their score. The tokenizer and lemmatizer used are part of the Stanford CoreNLP tagging tool (Toutanova et al., 2003). We then annotate for each question the class which corresponds to the passage retrieval part and the class which corresponds to the answer choice part, that is to say, the class that is necessary and sufficient to distinguish the correct item from the incorrect ones. When our baseline ranks the correct passage in first place, we automatically annotate this passage retrieval step as being of class 1. Similarly, when our baseline ranks the correct answer in first place –that is to say, the correct answer has strictly more words in common with the correct passage than all the other candidates–, we automatically annotate this answer choice step as being of class 1. The rest of the classes are manually annotated by two annotators (the authors of this paper) on all questions of the first document of each topic (documents 1, 5, 9 and 13) for the Main task, and on the first 7 documents of the Exams task, and a Cohen's Kappa score of inter-annotator agreement is calculated (Cohen, 1968).

We get a Cohen's Kappa of 0.81 on the passage retrieval subtask and 0.86 on the answer choice subtask for the Main dataset, on 51 questions annotated by both annotators, which denotes very high agreement. However, we have only a Cohen's Kappa of 0.49 on the passage retrieval subtask and 0.57 on the answer choice subtask for the Exams dataset, on 30 questions annotated by both annotators, which is a much lower agreement.

We observe (Tables 13 and 14) that the class distribution for the overall question answering task is different for Exams questions, even though the problem and format are the same. Exams contain significantly more questions of Tier 3 compared to Main. And among Tier 1 classes, they exhibit five times as much Class 2 phenomena as Class 1, whereas Main's first two classes are balanced, with even a slight bias toward simple pattern matching questions.

[2] In practice, we likely misestimate the difficulty of the passage retrieval subtask, because actual passage retrieval as performed by systems can benefit from including words of the right candidate answer, but likewise can be noisier as we add words of the wrong candidate answers. Nevertheless, a human reader can find most of the relevant passages without reading the corresponding provided candidate answers, so separating the task in those 2 subtasks is reasonable.

Class	PR (%)	AC (%)	QA (%)
1:Pattern matching	55	52	24
2:Relatedness	12	6	16
3:Structure	16	32	37
4:Concepts and Relations	4	0	2
5:Events and discourse	14	8	18
6:Visualization	0	2	2
7:?	0	0	0

Table 13: Class distribution for Main (docs 1, 5, 9 and 13) found by annotator 1

Class	PR (%)	AC (%)	QA (%)
1:Pattern matching	38	33	4
2:Relatedness	15	16	19
3:Structure	13	13	17
4:Concepts and Relations	0	2	2
5:Events and discourse	32	22	35
6:Visualization	4	13	17
7:?	0	0	0

Table 14: Class distribution for Exams (all docs) found by annotator 1

This confirms the hypothesis that the nature of questions in the Entrance Exams corpus is different from that of the questions in Main. Tests for entrance exams use more reformulations of the text (pertaining to Class 2 inferences) to test the english vocabulary of the student. The questions can also involve cognitive processes of higher order (Class 5 and more) which make heavy use of common-sense knowledge, as it is assumed to be naturally available to a human being.

The table 15 attempts to provide some insights on why our inter-agreement rate is much lower on the Exams task than on the Main task. We found that for more difficult problems, it can be confusing to pinpoint the class to which a phenomenon belongs. For example, we report that there seems to be confusions between Class 2 and Class 4 at the passage retrieval level. Class 2 introduces the notion of semantic relatedness and exterior knowledge can already at this level be captured through the resources employed, which makes it close to Class 4 in term of external world knowledge.

Nevertheless, the class distribution of Entrance Exams being skewed towards the upper Tier 2 and Tier 3, with respect to Main's distribution, we could expect the performance of systems which ran on both at QA4RME 2013 to be lower because the problem is overall harder. This is indeed the case, as seen in Table 16, all the more so as Entrance exams feature only 4 choices while Main features 5.

5. Conclusion

We described a hierarchical taxonomy of textual inferences. This taxonomy is designed with computational systems in mind, and encompasses tasks, techniques, tools and resources. As seen in the experiments, it can indeed make the contrasts within a dataset apparent, both in term of nature and overall difficulty of the task. From there we can guide further improvements on systems and choose the kind of problems we want to evaluate them on.

In future work, we plan to develop the specification of the annotation process of those classes, perform more fine-grained evaluations on systems and finally propose solutions to improve system performances on the higher classes of inference problems.

Annotators	Annotator 1		Annotator 2	
Annotators	PR	AC	PR	AC
1:Pattern matching	50	33	50	33
2:Relatedness	15	13	0	17
3:Structure	8	10	4	7
4:Concepts and Relations	0	3	19	27
5:Events and discourse	27	20	23	7
6:Visualization	0	20	4	10
7:?	0	0	0	0

Table 15: Class distributions from 2 annotators for Exams (docs 1 to 7)

Systems	Main (c@1)	Exams (c@1)
jucs	0.59	0.42
nara	0.33	0.22
limsi	0.28	0.22
Class	Main (%)	Exams (%)
1:Pattern matching	24	4
2:Relatedness	16	17
3:Structure	37	17
4:Concepts and Relations	2	0
5:Events and discourse	18	40
6:Visualization	2	17

Table 16: c@1 of systems at QA4MRE 2013 and question classes, for both tasks

6. References

Clark, P., Harrison, P., and Yao, X. (2012). An entailment-based approach to the QA4MRE challenge. In *CLEF (Online Working Notes/Labs/Workshop)*. Citeseer.

Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213.

Dagan, I., Glickman, O., and Magnini, B. (2006). The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190. Springer.

Dzikovska, M. O., Nielsen, R. D., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., and Dang, H. T. (2013). SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 263–274. Association for Computational Linguistics.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J., et al. (2010). Building Watson: An overview of the DeepQA project. *AI magazine*, 31(3):59–79.

Huang, H.-H., Chang, K.-C., and Chen, H.-H. (2013). Modeling human inference process for textual entailment recognition. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 446–450, Sofia, Bulgaria, August. Association for Computational Linguistics.

Levesque, H. J., Davis, E., and Morgenstern, L. (2011). The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*.

MacCartney, B. and Manning, C. D. (2007). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200. Association for Computational Linguistics.

McKoon, G. and Ratcliff, R. (1992). Inference during reading. *Psychological review*, 99(3):440.

Miller, G. A. (1995). WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Moldovan, D., Paşca, M., Harabagiu, S., and Surdeanu, M. (2003). Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems (TOIS)*, 21(2):133–154.

Moldovan, D., Clark, C., Harabagiu, S., and Hodges, D. (2007). Cogex: A semantically and contextually enriched logic prover for question answering. *Journal of Applied Logic*, 5(1):49–69.

Peñas, A. and Hovy, E. (2010). Filling knowledge gaps in text for machine reading. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 979–987. Association for Computational Linguistics.

Peñas, A., Hovy, E., Forner, P., Rodrigo, Á., Sutcliffe, R., and Morante, R. (2013). QA4MRE 2011-2013: Overview of question answering for machine reading evaluation. In *Information Access Evaluation. Multilinguality, Multimodality, and Visualization*, pages 303–320. Springer.

Poon, H. (2010). Markov logic in natural language processing: Theory, algorithms, and applications. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, page 3.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, pages 433–460.

Zwaan, R. A. and Radvansky, G. A. (1998). Situation models in language comprehension and memory. *Psychological bulletin*, 123(2):162.