Meaning as Context and Subsequence Analysis for Entailment

Daoud Clarke
Department of Informatics
University of Sussex
d.clarke@sussex.ac.uk

Abstract

Based on the idea that meaning is determined by context, we give a formal definition of entailment between two sentences in the form of a conditional probability on a measure space. We provide three practical implementations of this formalism: a bag of words comparison as a baseline, and two methods based on an analysing subsequences of the sentences, where by a "subsequence" y of a sequence x we mean that each symbol in y occurs in the same order in x, but possibly with intervening symbols. The second of these makes use of a large corpus to represent the meaning of each word by replacing it with the bag of document numbers in which it occurs.

Our results indicate that the subsequence method performs better than simple word matching, with the greatest benefit being in the information retrieval task, with approximately 7% increase in accuracy.

1 Introduction

In approaching the task of textual entailment, we were interested finding a formalism that would make explicit the assumption that meaning in language is determined by context. We were also looking for a way of describing *degrees* of entailment, rather than the strict true or false of logical implication. This approach is intended for cases where entailment is likely, but not certain; for some applications it may be useful to have a measure of this uncertainty. For the purpose of the Recognising Textual Entailment Challenge, the degree of entailment is used as a degree of confidence in order to rank our judgments.

Our formalism makes use of the concept of a *measure space*, a generalisation of a probability space which is not required to be normalised (we leave the formal definition to an appendix since it is not directly relevant to what follows).

We assume our sentences are taken from some finite alphabet A. Informally, each sentence or phrase $s \in A^*$ is associated with a set $\phi(s)$ which we interpret as the "set of contexts" of s. We then specify the "size" of these sets with a function μ. We leave open the exact method of determining these sets of contexts and sizes (i.e. different functions ϕ and μ), since different functions may be appropriate for different applications. Later we will define the specific functions that we have used in the second Recognising Textual Entailment Challenge.

The following definition forms the basis of our work in textual entailment recognition.

Definition 1. *Given a measure space $\langle X, \Sigma, \mu \rangle$, and a map $\phi : A^* \rightarrow \Sigma$, and given two sentences $s_1, s_2 \in A^*$, the degree to which s_1 entails s_2 is defined as*

$$\mathrm{Ent}(s_1, s_2) = \frac{\mu(\phi(s_1) \cap \phi(s_2))}{\mu(\phi(s_1))}$$

when $\phi(s_1)$ is not empty, and is left undefined otherwise.

Thus entailment takes the form of a *conditional probability* with respect to the measure spaces.

Glickman et al. also use conditional probability in their definition for entailment, in their entry to the first Recognising Textual Entailment Challenge (Dagan et al., 2005), and in (Glickman et al., 2005). The difference in our definitions are largely philosophical: they use the notion of "possible words" to describe the probability of sentences being true, whereas the intuition behind our definition is purely related to contexts. In addition they consider a sentence h to entail a sentence t if h increases the probability that t is true, $P(\mathrm{Tr}_h = 1|t) > P(\mathrm{Tr}_h = 1)$; they then use the conditional probability as a confidence in the entailment being true. In contrast, we consider the conditional probability to represent a *degree* of entailment, and there is no specified threshold at which entailment is considered to exist.

2 Implementations

For our entry to the Second Recognising Textual Entailment Challenge we decided to concentrate on simple lexical matching methods although we believe our definition to be more generally applicable. Specifically we decided to investigate the importance of *word order* in determining textual entailment. In one implementation we map a sequence of words to the set of *subsequences* of words, where the subsequence may include gaps; such methods have been used with success in text classification (Lodhi et al., 2002).

We describe here three implementations, the first of which is a baseline and was not entered into the challenge. In all cases the alphabet A is taken to be the set of words.

1. **Simple word matching.** In this implementation the function ϕ_1 simply maps a sentence to the bag (or multiset) of words in the sentence.

2. **Subsequence matching.** The second function ϕ_2 maps a sequence $s \in A^*$ to the bag of subsequences of s, where by a "subsequence" x of a sequence s we mean that each symbol in x occurs in the same order in s, but possibly with intervening symbols. For example, the sequence *a mercenary group* maps to the bag {*a, mercenary, group, a mercenary, a group, mercenary group, a mercenary group*}. A sequence s maps to a bag of $2^{|s|} - 1$ subsequences, where $|s|$ is the length of s.

3. **Subsequence matching with context replacement.** Our most complex implementation replaces each word w with a bag $\psi(w)$ of document identifiers (defined below) representing the contexts that w occurs in. We then define the function

$$\pi(w_1 w_2 \dots w_n) = \psi(w_1)\psi(w_2)\dots\psi(w_n),$$

mapping sequences to bags of sequences in D^* where D is the set of possible document identifiers. The function ϕ_3 is then defined in terms of ϕ_2 and π:

$$\phi_3(s) = \bigcup_{x \in \phi_2(s)} \pi(x),$$

where by $\bigcup$ we mean the bag-theoretic union. Thus ϕ_3 can be considered as mapping a sequence s to the bag of all sequences $d_1 d_2 \dots d_n$ of document identifiers such that there is a subsequence $w_1 w_2 \dots w_n$ of s where each word w_i occurs in context d_i.

The third implementation was intended to get closer to our intention of representing the meaning of words by the contexts that they occur in. Words are represented as the bag of documents that they occur in, along the lines of techniques used in information retrieval, such as latent semantic analysis (Deerwester et al., 1990); ϕ_3 then combines these vectors without discarding information about the order of words in the sequence.

In all the implementations, a set of stopwords (the 21 words occurring most frequently in the Gigaword corpus[1]) were removed before the analysis was performed. All words were converted to lowercase and all symbols were discarded.

We chose as the threshold value of entailment the mode of the values under consideration, so that exactly half the pairs were assigned "true" and half "false". No optimisation was performed on the development set, so results on this set should provide valid evidence for the behaviour of our system.

2.1 Corpus analysis

The third implementation makes use of the English Gigaword corpus to provide the function ψ. We

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

looked only at which documents the words occurred in the corpus, ignoring word order, thus modelling words as a "bag of documents" along the lines of techniques applied in information retrieval.

The function ψ was defined informally as follows. For each word (excluding stopwords) occurring more than ten times in the corpus we stored the bag of document identifiers corresponding to each occurrence of the word in the corpus. For words on which we had no data, each word was treated as if it had occurred in a single document by itself, as a simple form of smoothing.

2.2 Bag Theory and Measures

Bags (multisets) are useful for us in comparison to sets, as they retain information about the frequency of occurrence of elements. All our implementations use bags, but in order to use them with our definition of entailment we need to show how they can be considered to be measure spaces.

For any finite set S, clearly $\langle S, 2^S, o\rangle$ is a measure space, where o measures the cardinality of the set. We associate a set S_B with each bag B as follows:

$$S_B = \bigcup_{x \in B} \{x_1, x_2 \ldots x_{m(x)}\}$$

where x ranges over distinct elements in B, the subscripted elements are new distinct elements in S_B, and $m(x)$ is the multiplicity of x in B. Then $o(S_B)$ is the size of B, and we can thus consider bags as measure spaces by mapping them to their equivalent sets.

2.3 Computation

Our third implementation in particular presents computational difficulties since the number of elements in $\phi_3(x)$ is very large, even for moderately small sequences. This is in part due to the often very large number of occurrences of words throughout the Gigaword corpus, meaning large bags ψ for each word. However even ϕ_2 produces produces bags whose size is exponential in the length of the sequence.

To get around this problem we used a Monte-Carlo sampling technique to estimate the size of such sets. In order to estimate the intersection $\phi_3(s_1) \cap \phi_3(s_2)$ for two sequences s_1 and s_2 we repeatedly generated random subsequences of the same length u_1 and u_2 of each sentence, and calculated the size of the intersection $o(\pi(u_1) \cap \pi(u_2))$. The subsequences were made to be the same length since the intersection for sequences of different length would be empty, from the definition of π.

The mean m and variance v of this value was calculated intermittently and the calculation was terminated when the variance was less than a threshold, $v/m < t$, where we used a threshold value of $t = 0.01$, or a maximum number of a million cycles was reached. The resulting value of the mean, $\mathrm{MC}(s_1, s_2)$ from the Monte-Carlo technique was used to estimate the real value of $\mathrm{Ent}(s_1, s_2)$ by

$$\frac{o(\phi_3(s_1) \cap \phi_3(s_2))}{o(\phi_3(s_1))} \simeq \frac{\mathrm{MC}(s_1, s_2)}{\mathrm{MC}(s_1, s_1)}.$$

This assumes that $o(\phi_3(s_1) \cap \phi_3(s_2))$ is proportional to $\mathrm{MC}(s_1, s_2)$. This can be justified by considering a random element of $\phi_3(s_1)$ and a random element of $\phi_3(s_2)$. The probability that these two are the same will be proportional to $o(\phi_3(s_1) \cap \phi_3(s_2))$. The Monte-Carlo technique generalises this by looking for matching elements in *subsets* of $\phi_3(s_1)$ and $\phi_3(s_2)$. These are, however, clearly not random subsets, and thus there may be some unknown bias in this approximation. The Monte-Carlo technique is used twice to provide the appropriate renormalisation factor.

To provide a fair comparison, the same technique was applied to the other functions ϕ, even though this was strictly unnecessary in computational terms.

2.4 Error Analysis

The convention in most scientific literature is to use the standard deviation of the mean of a sample as an estimate of the error in the measured mean. We will show in detail how this can be applied to estimating the error in our measured values of accuracy in the Recognising Textual Entailment challenge.

For any entailment-judging system s, let J_s denote the random variable which takes the value 1 when the system correctly judges an entailment, and 0 when it doesn't. We assume that J_s is distributed binomially, $J_s \sim \mathrm{Bin}(n, a_s)$ where n is the number of entailment pairs and a_s is the accuracy of system s. This assumption seems justified since we may imagine that the pairs are randomly sampled from

Run	ϕ	Acc.	Av. Prec.	IE	IR	QA	SUM	Dev. Acc
Word Matching	ϕ_1	0.53	0.56	0.46	0.49	0.59	0.59	0.57
Subsequence Matching	ϕ_2	0.55	0.53	0.49	0.56	0.54	0.60	0.57
Substr. + Corpus Occ.	ϕ_3	0.53	0.53	0.50	0.50	0.53	0.59	0.55

Table 1: Estimates of overall accuracy and average precision, and accuracy results for each of the subtasks (Information Extraction, Information Retrieval, Question Answering and Summarisation) on the test set, and accuracy for the development set. Results are reported for a baseline of simple word matching and the two entered runs: subsequence matching and subsequences combined with corpus occurrences. Error for accuracy values on the test and development sets is approximately 2%, and on the subtasks, 4%.

some infinite set of pairs. Then, given an observation of c correct judgments out of the n trials, our best estimate of a_s is clearly c/n. Using this estimate[2], we may expect a variance in subsequent trials of $a_s n(1-a_s) \simeq c(1-c/n)$. We may thus estimate the error $\sigma(a_s)$ in the value of a_s as a fraction of the standard deviation:

$$\sigma(a_s) = \frac{1}{n}\sqrt{c(1-c/n)}$$

For example, for a sample of $n = 800$, getting exactly $c = 400$ judgments right gives an estimate of 0.0177 for the error in the accuracy, which we write as $a_s = 0.50 \pm 0.02$, since the convention is to quote error to a single significant figure. The error does not vary much as a_s varies: for a value of $c = 600$ correct judgments, $\sigma(a_s) = 0.0153$, so $a_s = 0.75 \pm 0.02$. Thus, for the range of values likely to be of interest to us in the challenge, the error will always be 2% for the entire test set. For the smaller sample sizes of 200 pairs in each of the subtasks, error is between 3% and 4%.

To test for significance of a result, we estimate the probability that our system could have gained the same number of correct judgments or better by assigning true or false at random, that is, its true value of a_s is 0.5. For 800 trials, the probability of scoring 434 or more is less than 0.01 (given a true accuracy of 0.5), so a measured accuracy of $434/800 \simeq 0.54$ is significant at the 1% level. For the sub-tasks, only an accuracy of 0.59 is of the same significance.

[2]Our uncertainty about the true value of a_s is not terribly important, since as we will see, the error does not change greatly for different values of a_s within the range we are interested in.

3 Results and Analysis

Results for the two submitted entries and a baseline are shown in table 1.[3] It was found that for the functions ϕ we considered, $\mathrm{Ent}(h,t)$ was a better measure of entailment than $\mathrm{Ent}(t,h)$ (for text and hypothesis sentences t and h respectively), and this was what was used in the submitted runs. We hypothesise that this is because the particular functions ϕ that we considered do not provide an estimate of the frequency of occurrence of a sequence: longer sequences have higher measures, whereas we would obviously expect longer sequences to occur less often. This is something we hope to investigate further in future work.

The results seem to show that taking into account word order does have an impact on system performance. Combining the results of the development and the test set, the simplest system (function ϕ_1) has an accuracy of 0.55 ± 0.01, whereas the system which uses subsequence matching (function ϕ_2) has an accuracy of 0.56 ± 0.01.

The main benefit seems to have been in the information retrieval (IR) task, in which word order provided a 7% improvement over simple word matching in the test set. (Doucet and Ahonen-Myka (2004) describe a system that uses multi-word expressions to improve retrieval performance; according to them most information retrieval systems do not account for word order in a document. However, Google clearly makes use of word order, as it returns a significantly different set of documents for the query *order word* as compared to *word order*.) It could be that word order is especially useful in this task as it does not require any higher level reason-

[3]Note: ϕ_2 corresponds to submitted file "run2.txt" and ϕ_3 corresponds to submitted file "run1.txt".

ing, as compared to the question answering task, for example, where word order may in fact be misleading — subsequence matching led to a reduction in accuracy of around 5% as compared to simple word matching.

The summarisation (SUM) task was the best performing for all systems and the most difficult task seems to be information extraction (IE), presumably because the latter requires higher level reasoning in order to determine whether entailment does not exist. Many pairs for which entailment does not exist in the summarisation task can easily be detected by the appearance of unrelated words in the test sentence, for example, for the text-hypothesis pair

- *Baldwin graduated from Butler University in 1995, and he served his alma mater as an assistant coach from 1994-99.*
- *Baldwin dismissed by Butler University.*

(pair ID 605 in the test set) entailment is easily rejected by noting that the word "dismissed" in the hypothesis sentence is unrelated to any word in the text sentence.

Strangely, the inclusion of context information seems to have been detrimental to the performance of the system. This is unexpected given previous work, for example (Jijkoun and de Rijke, 2006) where use of lexical similarity measures is shown to improve the performance of their system. We can only put this down, again, to the non-probabilistic nature of the function ϕ_3. Replacing words by their sets of occurrence would increase the emphasis on longer strings: if each word is replaced by n document identifiers then a subsequence of length l would have a size of n^l.

The average precision values seem to indicate also that our systems are not very good at ranking entailments, since straightforward word matching performs the best according to this measure.

There seems to be a consistent difference in performance between the test and development sets, though not beyond what we would expect given the sample size. The most significant difference was for simple word matching, which would imply that simple techniques would expect to do less favourably in the test set compared to what could be expected.

4 Conclusions and Further Work

We have described a new measure-theoretic framework for textual entailment, and three implementations. These show that making use of word order is beneficial in recognising entailment, especially in the information retrieval task.

Our system seems to have suffered from making use of non-probabilistic measures, hence in the future we intend to concentrate on new functions ϕ which more closely model the probability of occurrence of a sequence.

Acknowledgments

I wish to thank my supervisor David Weir, the University of Sussex for funding my research and an anonymous reviewer for comments.

Appendix — Definition of Measure Space

Definition 2. *A measure space is a triple $\langle X, \Sigma, \mu \rangle$ such that Σ satisfies:*

- *the empty set $\emptyset$ is in Σ;*
- *if A is in Σ then its complement $X - A$ is in Σ;*
- *if E is a countable collection of sets in Σ then the union of all the sets in E is in Σ*

(i.e. Σ is a sigma algebra *over X), and μ is a function from Σ to the non-negative real numbers*[4] *satisfying $\mu(\emptyset) = 0$ and*

$$\mu\Big(\bigcup_{A \in E} A\Big) = \sum_{A \in E} \mu(A),$$

where E is a countable set of disjoint sets in Σ.

References

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *The PASCAL Challenges Workshop on Recognising Textual Entailment, 2005*.

Scott Deerwester, Susan Dumais, George Furnas, Thomas Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

[4] The normal definition of measure allows infinite values for the function μ, we disallow this so that the conditional probability is well defined.

Antoine Doucet and Helena Ahonen-Myka. 2004. Non-contiguous word sequences for information retrieval. In *Proceedings of ACL-2004, Workshop on Multiword Expressions: Integrating Processing, Barcelona, Spain, July 21-26, 2004*, pages 88–95.

Oren Glickman, Ido Dagan, and Moshe Koppel. 2005. A probabilistic classification approach for lexical textual entailment. In *Proceedings of AAAI 2005*, pages 1050–1055.

Valentin Jijkoun and Maarten de Rijke. 2006. Recognizing textual entailment: Is lexical similarity enough? In I. Dagan, F. Dalche, J. Quinonero Candela, and B. Magnini, editors, *Evaluating Predictive Uncertainty, Textual Entailment and Object Recognition Systems*, LNAI. Springer Verlag.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. 2002. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444.