IT3105 Project III: Recognizing Textual Entailment

Part I

Erwin Marsi
IT-VEST 313
emarsi@idi.ntnu.no

Logic and Language Technology Group, Intelligent Systems, IDI, NTNU

Plan for today's lecture

1. Introduction
2. Recognizing Textual Entailment
3. Break (15 minutes)
4. Lexical pattern matching
5. Epilog

Introduction

- Easy for computers is often hard for us: calculating the square root of 6745734692?
- But easy for us is often hard for computers
- Goes for most AI problems
- Certainly for natural language
- How is it possible that after decades of research by thousands of bright researchers with exponentially growing computational power, computers still can't do what any normal toddler can: *learn a language?*

Introduction

- Admittedly progress has been made in computational treatment of language
- in fields such as Natural Language Understanding, Computational Linguistic, and Natural Language Processing
- Computers are fairly good at dealing with structural aspects of language
  - how complex words are composed from smaller parts (morphology)
  - how these words are combined into a grammatical sentence (syntax)
- But understanding the *meaning* of language is still a very hard problem...

The logic approach to semantics

- Long tradition of logic models for the semantics and pragmatics of language
- Models are elegant and insightful, but of limited use in practice
  1. insufficient *coverage*
  2. lack of *robustness*
  3. the *knowledge acquisition bottleneck*

Problem 1: Insufficient coverage

- Most models cover only the well-organized, but rather rare aspects of language
- For example: negation, quantification, modality, scope, etc.
- But real language is rather disorganized and messy
- Not many donkey-sentences in real language use: "*Every farmer who owns a donkey beats it.*"

Problem 2: Lack of robustness

- Most formal models are too brittle
- Input is assumed to be perfectly well-formed, standard language
- Small deviations bring the whole model down
- For example: a spelling error, a missing or unknown word, an idiomatic expression, an unfinished sentence
- Robust models exhibit graceful degradation: continue to operate properly in the event of one or more the failures.

Problem 3: Knowledge Acquisition Bottleneck

- Linguistic theory tends to be language-specific
- A grammer for English does not work for Arabic or Chinese
- For each language, you will need to find/create the required linguistic theory and derive/implement a computational model
- Infeasible in practice due to the huge efforts and costs involved

Empirical methods in NLP

- At the end of the last century, paradigm for NLP shifted from rational, knowledge based approaches to **empirical methods**
- Meaning: data-driven methods relying on statistics and machine learning
- Computers learn language processing directly from large collections of examples
- Mostly supervised machine learning, but unsupervised learning is gaining momentum
- Empirical methods have proven to be way more effective at solving real-world language processing tasks

Example: Machine Translation

- **Machine Translation** (MT): automatically translating one language into another
- In 70's and 80's MT was limited to a handful language pairs due to the enormous costs of implementing a full fledged system for a given language pair
- Today we have online services (Google Translate) providing translation for over 4000 language pairs
- Even though translation quality is often far from perfect, the quantity and speed alone is impressive

Empirical semantics

- This project is about applying empirical methods to semantics, or understanding the meaning of language
- In the form of very concrete task: Recognizing Textual Entailment

Textual entailment

- **Textual entailment** is defined as a directional relation between two text fragments, termed T - the entailing *text*, and H - the entailed *hypothesis*
- T entails H if, typically, a human reading T would infer that H is most likely true
- Somewhat informal definition assumes common human understanding of language as well as common background knowledge

Entailment examples

Id	Text	Hypothesis	Entails
1	The drugs that slow down or halt Alzheimer's disease work best the earlier you administer them.	Alzheimer's disease is treated using drugs.	YES
2	Drew Walker, NHS Tayside's public health director, said: "It is important to stress that this is not a confirmed case of rabies."	A case of rabies was confirmed.	NO
3	Yoko Ono unveiled a bronze statue of her late husband, John Lennon, to complete the official renaming of England's Liverpool Airport as Liverpool John Lennon Airport.	Yoko Ono is John Lennon's widow.	YES

Entailment examples

Id	Text	Hypothesis	Entails
4	Arabic, for example, is used densely across North Africa and from the Eastern Mediterranean to the Philippines, as the key language of the Arab world and the primary vehicle of Islam.	Arabic is the primary language of the Philippines.	NO
5	About two weeks before the trial started, I was in Shapiro's office in Century City.	Shapiro works in Century City.	YES

Entailment examples

Id	Text	Hypothesis	Entails
6	Meanwhile, in his first interview to a Western print publication since his election as president of Iran earlier this year, Ahmadinejad attacked the "threat" to bring the issue of Iran's nuclear activity to the UN Security Council by the US, France, Britain and Germany.	Ahmadinejad is a citizen of Iran.	YES

Directional

- Entailment relation is directional
- H must be entailed by T, but the T does need not be entailed from H hypothesis.
- Bidirectional entailment means the T and H are **paraphrases**
- Example: "*John is married to Mary*" and "*Mary is married to John*"

Common knowledge

- Definition of entailment allows presupposition of **common knowledge**
- Such as: a company has a CEO, a CEO is an employee of the company, an employee is a person, etc.

Entailment is different from truth

- H must be supported by T
- Even if H is evidently true, entailment can still be false, because T does not support it
- Example: "*Paris is the capitol of France.*"

Entailment is probabilistic in nature

- Textual entailment differs from strict logical entailment
- If entailment is very likely, it is judged as YES
- Compare example 5

Recognizing Textual Entailment

- **Recognizing Textual Entailment** (RTE) is the task of deciding, given T and H, whether T entails H.
- Carried out fully automatically by a computer program
- Without any human intervention

Contradicts vs. unsupported

- The class of NO entailment can be divided in two subclasses:
  - H is **unsupported** by T
  - H is **contradicted** by T
- Some extension to the RTE task formulate it as a three-way task
- Many systems have a module specialized in recognizing contradictions

Motivation for RTE

- RTE is interesting and worthwhile, from both scientific and applied points of view
- Allows empirical verification and comparison of very different approaches to semantics
- Stated at a very concrete level (text), so makes no theoretical assumptions
- Supports anything from low level pattern matching...
- combinations of deep logical analysis, huge knowledge bases and reasoning

Motivation for RTE

- RTE data sets are randomly sampled from real text (newspapers, manuals, webpages, . . .)
- Thus no artificial examples
- Examples are representative of everyday language use

Applications of RTE

- With natural language, we can say the same thing in many different ways
- Major obstacle for Natural Language Processing systems

RTE and Question Answering

- **Question Answering** (QA): given a question, automatically extract correct answers from a collection of documents
- Example Q: "*Who bought Overture?*"
- QA system analyses question
- Creates answer pattern: "*X bought Overture*"
- Searches in documents for text fitting the answer pattern
- However, the answer "*Yahoo bought Overture*" may not occur literally
- Instead QA system finds "*Overtures acquisition by Yahoo*"
- QA system now calls RTE module to verify that "*Overtures acquisition by Yahoo*" entails "*Yahoo bought Overture*"

Applications of RTE

- Similar arguments can be made for many other NLP applications,
- Including Information Retrieval (IR), Information Extraction (IE), Automatic Summarization and Machine Translation (MT)
- Proponents of textual entailment belief that most NLP tasks can be reduced to RTE
- To what extent this is true remains to be seen...

The RTE challenges

- **The Recognizing Textual Entailment Challenge** is a competition
- Held every year since 2004
- Common theme: participants need to build RTE system
- Task definition and data evolved over time
- At start, development data is released
- Collection of T-H pairs annotated by human judges for true/false entailment.
- Can be used by participants to develop their systems, to estimate accuracy, sometimes even to train or fine tune systems

The RTE challenges

- At the end, participants receive a test data set
- Blind: human entailment judgements are missing
- This prevents cheating, like tuning your system on the test data
- Participants run their system on the test data and return output to organizers
- Organizers then compute accuracy scores by comparing each system's predictions to the true values
- Results get published

Development of accuracy scores over time

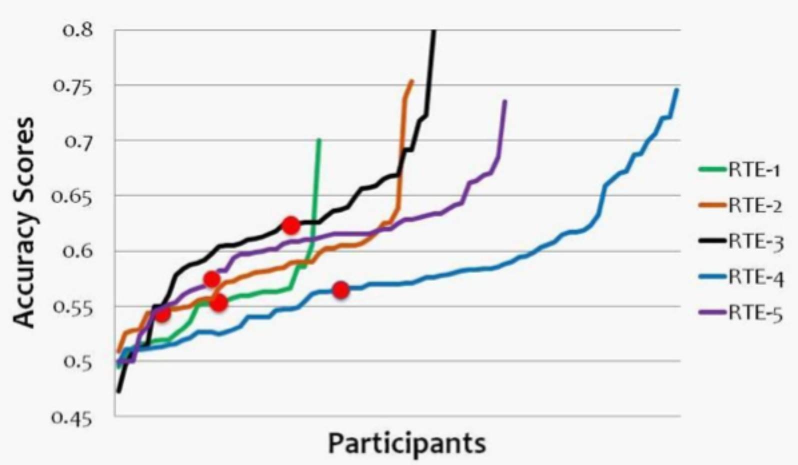

Figure: Accuracy scores per RTE challenge (from the RTE tutorial by Mark Sammons, Idan Szpektor V.G. Vinod Vydiswaran)

- Part IV of the curent project is modelled after the RTE challenge
- You build your system using the development data
- Blind test data will be released two days before the end of the project
- Accuracy scores on the test set determine credits awarded for Part IV
- No need to worry if your system's accuracy is far from 1.0
- Remember that average system performance in the RTE challenges ranges from 0.55 to 0.65

Word matching

- Assumption underlying most RTE approaches:
  if H is similar to T, then entailment is likely
- Requires matching or aligning H to parts of T
- Most basic method:
  - count no of shared words
  - set a threshold above which entailment is true
- Problem: the longer H is, the more words it will likely share with T
- Solution: normalize for sentence length by dividing by no of words in H

Word matching

$$Word\ Match = \frac{\#words\ in\ both\ H\ and\ T}{\#words\ in\ H}$$

- Determine optimal threshold by calculating accuracy on dev data for different thresholds
- Gives baseline accuracy: more complicated approaches must beat the baseline score

Lemma matching

- Word matching has obvious shortcomings

Example

T: He is looking out of the windows

H: He looks out of a window

?: YES

- Failed word matches:
  - *looks* does not match *looking*
  - *window* does not match *windows*

Lemma matching

- **Lemma**: underlying form of surface form of words
- many dictionaries are lemma-based

Example

Surface forms: *to look, looks, looking, looked*
Underlying lemma: *look*

Surface forms: *window windows*
Underlying lemma: *window*

Lemma matching

- Converting words to lemmas improves match

Example

T: *he be look out of the window*
H: *he look out of a window*
?: YES

- Automatic lemmatization can be hard
- Fortunately lemmas are available from **preprocessed** dev data
- Define Lemma Match measure and determine threshold, analogously to Word Match

Part-of-speech mapping

- Word matching has more shortcomings. . .

Example

T: *The lead singer left to wave at us*

H: *The left wave threw lead at us*

?: NO

- 6 out of 7 H words match a T word
- But some matching words have completely different meaning!
  1. *lead* in the sense of "first" does not match *lead* in the sense of "metal"
  2. *left* in the sense of "going aways" does not match *left* in the sense "opposite of right"
  3. *wave* in the sense of "greeting" does not match *wave* in the sense of "movement in fluid"

Part-of-speech mapping

- **Homographs**: words with identical spelling but different meanings
- Homographs can often be distinguished by their word class
- **Part of speech** (POS): the lexical category of a word
- Some major words categories in English are
  - Noun (N) as in *car*
  - Verb (V) as in *drive*
  - Adjective (A) as in *blue*
  - Adverb (Adv) as in *quickly*)

Part-of-speech mapping

- **Part-of-speech tagging**: process of labeling words according to a predefined set of POS tags

Example

T: *The/Det lead/N singer/N left/V to/Aux wave/V at/Prep us/Pro*
H: *The/Det left/A wave/N threw/V lead/N at/Prep us/Pro*
?: NO

- Matching on word + POS tag partly solves our problem
  - *wave/V* does no longer match *wave/N*
  - *left/V* does no longer match *left/A*
- Not completely though: *lead/N* still matches *lead/N*

Part-of-speech mapping

- POS tagging with high accuracy is not easy
- Fortunately, POS tags are available in preprocessed RTE data
- Define Word + POS Match measure and determine threshold, analogously to Word Match
- Note: no need to understand what each POS tag means
- Warning: homographs are infrequent, so do not expect significant improvements

Inverse Document Frequency

- Some words matter more than others
- Search engine filter most frequent stopwords from query
- Remaining query terms are usually weighted
- TF*IDF (Term Frequency times Inverse Document Frequency) is very common weighting scheme from Information Retrieval
- We ignore the TF part here

Inverse Document Frequency

- IDF is defined over a collection of documents
- $DF(w)$ = number of docs containing word w
- $IDF = 1/DF$
- Interpretation:
  - If w occurs in many documents, then $DF(w)$ is high, thus $IDF(w)$ low, meaning it is not a good for discriminating docs
  - If w occurs in few documents, then $DF(w)$ is low, thus $IDF(w)$ is high, meaning it is a good indicator for specific docs
- IDF weighting of words has generally been found to improve search results

Inverse Document Frequency

- IDF weighting can also be applied to word matching for RTE
- Define document collection as all T and H in the data set
- Compute IDF values for all words in this collection

$$IDF\ Word\ Match = \frac{\sum_{w \in T \cap H} IDF(w)}{\sum_{w \in H} IDF(w)}$$

BLEU algorithm

- BLEU (BiLingual Evaluation Understudy) algorithm stems from Machine Translation
- evaluation of translation output by humans is time-consuming and expensive
- BLEU performs an automatic evaluation that correlates reasonably well with human judgements
- Allows quick evaluation and optimization of MT systems
- BLEU essentially measures overlap between a system translation and human reference translations

BLEU algorithm

- **N-gram**: a subsequence of n items from a given sequence (for $n >= 1$)
- A word n-gram is thus a subsequence of n words from a sentence

Example

Sentence: *There is a cat on the mat*

Bigrams:	*There is*	Trigrams:	*There is a*
	is a		*is a cat*
	a cat		*a cat on*
	cat on		*cat on the*
	on the		*on the mat*
	the mat		

BLEU algorithm

- list of all 1-grams (*unigrams*) is simply the list of all words
- n-grams can occur more than once in a sentence, so we often talk about **n-gram counts**

BLEU algorithm

Example (translation candidates and reference translations)

C1: *It is a guide to action which ensures that the military always obeys the commands of the party.*
C2: *It is to insure the troops forever hearing the activity*

R1: *It is a guide to action that ensures that the military will forever heed Party commands.*
R2: *It is the guiding principle which guarantees the military forces always being under the command of the Party.*
R3: *It is the practical guide for the army always to heed the directions of the party.*

- Observation: good translation C1 shares more n-grams (words and phrases) with references than bad translation C2

BLEU algorithm

- The Word Similarity measure described earlier is in fact precision on unigrams

$$Prec_1 = \frac{|w \in Cand \cap Refs|}{|w \in Cand|}$$

- Can be generalized to precision on n-grams

$$Prec_n = \frac{|ngram \in Cand \cap Refs|}{|ngram \in Cand|}$$

- N-gram precision tells how many of the n-grams in the candidate can be found in *any* of the references

BLEU algorithm

- 1-gram precision indicates **adequacy**: to what extent the translation conveys the information in the source sentence
- higher n-gram precision indicates **fluency**: to what extent the translation is a well-formed expression in the target language
- To capture both aspects, we take an average over the n-gram precisions
- However, the higher the n-gram, the less likely a match with the reference, thus the lower the n-gram precision
- Arithmetic mean is therefore dominated by unigram precision, that is by adequacy, at the cost of fluency

BLEU algorithm

- Correction: take the *geometric mean* over the n-gram precisions

$$mean\ precision = \sqrt[n]{prec_1 \times prec_2 \times \ldots \times prec_n} \tag{1}$$

- which is equivalent to taking the average over the *logs* of the n-gram precisions

$$mean\ precision = exp\left[\frac{1}{N}\sum_{n=1}^{N} log\ \ prec_n\right] \tag{2}$$

BLEU algorithm

$$BLEU = mean\ precision \times BP$$

- Brevity Penalty (BP) penalizes translations that are shorter than the reference translations
- For example, translation candidate "the" is likely to have a BLEU score of 1.0!
- A way to compensate for a lack of recall
- Implementation details of BP are irrelevant for our purposes
- N-gram precision for $n > 4$ turns out be negligible, so by default BLEU score is computed for n up to 4.

BLEU algorithm

$$BLEU = mean\ precision \times BP$$

- Brevity Penalty (BP) penalizes translations that are shorter than the reference translations
- For example, translation candidate "the" is likely to have a BLEU score of 1.0!
- A way to compensate for a lack of recall
- Implementation details of BP are irrelevant for our purposes
- N-gram precision for $n > 4$ turns out be negligible, so by default BLEU score is computed for n up to 4.

BLEU algorithm

- BLEU has been used in RTE measure the match of H (candidate translation) to T (reference translation)
- However, penalty for brevity makes no sense, because it does not matter that H is shorter than T
- Solution: remove BP factor

BLEU algorithm

- BLUE is designed to work with *multiple* human reference translations
- However, there is just one reference (T) in RTE
- Therefore, n-grams are less likely to match and n-gram precision is often zero
- But then mean precision is zero, and BLEU score is zero!
- Solution: return to arithmetic mean
- Appears to work better in practice, even though it reduces the influence of fluency

$$Modified\ BLEU = \frac{1}{N}\sum_{n=1}^{N} prec_n$$

BLEU algorithm

- Details of implementation:
  - Lower-case all words first
  - Determine threshold for Modified BLUE score on development data

How to go on from here

- Read the project description
- Read the first part of the lecture notes
- Download and get familiar with the development data, both the original and the preprocessed version
- Start with implementing the most simple word matching approach
- Evaluate your system's output using the evaluation script from the website
- Continue with the more advanced methods of matching in Part I of the project
- Next lecture is in room F6 on Monday, 7th on november, 17.15-19.00 hrs