Natural Language Inference

Compositional Entailment

Birgit Schwarz

Saarland University

July 11, 2011

Introduction

Ed didn’t manage to remember to open the door.

Introduction

Ed didn’t manage to remember to open the door.

Did Ed open the door?

Introduction

- build a theory of compositional entailment
- Principle of Compositionality:
  The meaning of a compound expression is a function of the meanings of its parts.
- if two expressions differ by a single atomic edit, then the entailment relation between them depends on:
  - lexical entailment relation generated by edit
  - effect context of the expression has on the entailment relation
- atomic edits: substitution(SUBS), deletion (DEL), insertion (INS)

Lexical entailment relations

x	compound linguistic expression red car
$e(x)$	result of applying an atomic edit e to x SUB(*car, convertible*) $\Rightarrow$ red convertible
$\beta(e)$	lexical entailment relation generated by e car $\sqsupset$ convertible
$\beta(x, e(x))$	entailment relation between x and $e(x)$, depending on $\beta(e)$ and context of x red car $\sqsupset$ red convertible

Lexical entailment relations

x	compound linguistic expression red car
$e(x)$	result of applying an atomic edit e to x SUB(*car, convertible*) $\Rightarrow$ red convertible
$\beta(e)$	lexical entailment relation generated by e car $\sqsupset$ convertible
$\beta(x, e(x))$	entailment relation between x and $e(x)$, depending on $\beta(e)$ and context of x red car $\sqsupset$ red convertible

How can these entailment relations be computed?

Lexical Entailment

- lexical entailment relation generated by a substitution equals relation between the two terms: $\beta(\text{SUB}(a, b)) = \beta(a, b)$

Hyperonym:	car	$\sqsupset$	convertible
Synonym:	forbid	$\equiv$	prohibit
Hyponym:	crow	$\sqsubset$	bird
Antonym:	warm	$\mid$	cold

- relations can be acquired via WordNet

Entailments and Semantic Composition

- so far, we can determine entailment relations between isolated terms

hug $\sqsubset$ touch
French | German

- but how do these entailment relations behave in a context?

doesn't hug ? doesn't touch
not French ? not German

Entailments and Semantic Composition

- so far, we can determine entailment relations between isolated terms

hug $\sqsubset$ touch
French | German

- but how do these entailment relations behave in a context?

doesn't hug $\sqsupset$ doesn't touch
not French $\smile$ not German

How is a relation projected through a context?

Monotonicity Calculus

- developed by Sánchez Valencia in 1995
- explains impact of semantic composition on $\equiv, \sqsubset, \sqsupset$, and #
- three monotonicity classes:
  - UP projects entailment relations without change:
    parrot $\sqsubset$ bird $\Rightarrow$ parrots talk $\sqsubset$ some birds talk
  - DOWN swaps $\sqsubset$ and $\sqsupset$
    carp $\sqsubset$ fish $\Rightarrow$ no carp talk $\sqsupset$ no fish talk
  - NON projects $\sqsubset$ and $\sqsupset$ as #
    human $\sqsubset$ animal $\Rightarrow$ most humans talk # most animals talk
- lacks handling of exclusion relations $\hat{}, |$, and $\smile$

Projectivity

- generalize the concept of monotonicity to a concept of projectivity
- specify how an entailment relation is projected through a semantic composition tree
- Principle of Compositionality:
  The entailments of a compound expression are a function of the entailments of its parts

Projectivity of Logical Connectives: Negation

- projects $\equiv$ and # without change
- is downward monotonic, therefore swaps $\sqsubset$ and $\sqsupset$
- swaps $|$ and $\smile$

happy	$\equiv$	glad	$\Rightarrow$	not happy	$\equiv$	not glad
kiss	$\sqsubset$	touch	$\Rightarrow$	didn't kiss	$\sqsupset$	didn't touch
human	$\hat{}$	nonhuman	$\Rightarrow$	not human	$\hat{}$	not nonhuman
French	$\mid$	German	$\Rightarrow$	not French	$\smile$	not German
swimming	$\#$	hungry	$\Rightarrow$	not swimming	$\#$	not hungry

Projectivity of Logical Connectives: Conjunction

- “and” is upward monotone
- projects both $\hat{}$ and $|$ as $|$
- intersective modification (by adjectives, adverbs) has the same projectivity

convertible	$\sqsubset$	car	$\Rightarrow$	red convertible	$\sqsubset$	red car
human	$\hat{}$	nonhuman	$\Rightarrow$	living human	$\mid$	living nonhuman
French	$\mid$	Spanish	$\Rightarrow$	French wine	$\mid$	Spanish wine

Projectivity of Logical Connectives: Disjunction

- is upward monotone like conjunction
- unlike conjunction, projects both $\hat{}$ and $\smile$ as $\smile$ and projects $|$ as $\#$

waltzed	$\sqsubset$	danced	$\Rightarrow$	waltzed or sang	$\sqsubset$	danced or sang
human	$\hat{}$	nonhuman	$\Rightarrow$	human or equine	$\mid$	nonhuman or equine
red	$\mid$	blue	$\Rightarrow$	red or yellow	$\mid$	blue or yellow

Projectivity of Logical Connectives: Conditionals

- the antecedent of a conditional is downward-monotone
- the consequent is upward-monotone
- the antecedent projects both $\hat{}$ and $|$ as $\#$
- the consequent projects both $\hat{}$ and $|$ as $|$

If he drinks tequila, he feels nauseous	$\sqsupset$	If he drinks liquor, he feels nauseous
If he drinks tequila, he feels nauseous	$\sqsubset$	If he drinks tequila, he feels sick
If it's sunny, we surf	$\#$	If it's not sunny, we surf
If it's sunny, we surf	$|$	If it's sunny, we don't surf

Projectivity of Quantifiers

- all quantifiers project $\equiv$ and $\#$ as without change
- $|$ is projected as $\#$

dog	$\sqsubset$	animal	$\Rightarrow$	some dogs	$\sqsubset$	some animals
car	$\sqsupset$	convertible	$\Rightarrow$	no car	$\sqsubset$	no convertible
human	$\hat{}$	nonhuman	$\Rightarrow$	most humans	$\#$	most nonhumans
animal	$\smile$	non-ape	$\Rightarrow$	ex. one animal	$\#$	ex. one non-ape

Projectivity of Verbs

- most verbs are upward-monotone
- many verbs project $\hat{}$, $\smile$, and $|$ as $\#$

humans	$\hat{}$	nonhumans	$\Rightarrow$	eats humans	$\#$	eats nonhumans
cats	$\vert$	dogs	$\Rightarrow$	eats cats	$\#$	eats dogs

Projectivity of Verbs

- most verbs are upward-monotone
- many verbs project $\hat{}$, $\smile$, and $|$ as $\#$

humans	$\hat{}$	nonhumans	$\Rightarrow$	eats humans	$\#$	eats nonhumans
cats	$\vert$	dogs	$\Rightarrow$	eats cats	$\#$	eats dogs

- but:

Tom forgot to close the door	$\models$	The door isn't closed
Tom didn't forget to close the door	$\models$	The door is closed
Tom forgot that the door was closed	$\models$	The door is closed
Tom didn't forget that the door was closed	$\models$	The door is closed

Definition

Factive verbs

- carry same implication in both positive and negative contexts
- *admit that, forget that, believe that*...
- rather presuppose than entail truth of their complements, therefore not affected by negation

Implicative verbs

- implication depends on context
- *manage to, forget to, permit to, fail to, force to*...
- entail, rather than presuppose truth of their complements

Implication Signatures

- developed by Nairn et al. (2006)
- signatures model the directions of implications regarding the complements of verbs
  - positive (+), negative(-), null(○)

manage to (+ / -)

managed to escape	⇒	escaped
didn't manage to escape	⇒	didn't escape

refuse to (- / ○)

refused to dance	⇒	didn't dance
didn't refuse to dance	⇒	unclear

Implication Signatures: Deletion and Insertion of Implicatives

signature	$\beta(\mathit{DEL}(\cdot))$	$\beta(\mathit{INS}(\cdot))$			
$+/-$	$\equiv$	$\equiv$	he managed to escape	$\equiv$	he escaped
$-/+$	$\hat{}$	$\hat{}$	he failed to pay	$\hat{}$	he paid
$\circ/-$	$\sqsupset$	$\sqsubset$	he was permitted to live	$\sqsupset$	he lived
$-/\circ$	$\mid$	$\mid$	he refused to fight	$\mid$	he fought
$+/\circ$	$\sqsubset$	$\sqsupset$	he was forced to sell	$\sqsubset$	he sold
$\circ/+$	$\smile$	$\smile$	he hesitated to ask	$\smile$	he asked

Implication Signatures: Deletion and Insertion of Factives

signature	$\beta(DEL(\cdot))$	$\beta(INS(\cdot))$			
$+/+$	$\sqsubset$	**X**	he admitted that he knew	$\sqsubset$	he knew
$-/-$	$\mid$	**X**	he pretended he was sick	$\mid$	he was sick

Implication Signatures: Projectivity

Translating signatures into projectivity relations:

			projectivity						
signature	example	monotonicity	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$+ / -$	*manage to*	UP	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$+ / \circ$	*force to*	UP	$\equiv$	$\sqsubset$	$\sqsupset$	$\mid$	$\mid$	$\#$	$\#$
$\circ / -$	*permit to*	UP	$\equiv$	$\sqsubset$	$\sqsupset$	$\smile$	$\#$	$\smile$	$\#$
$- / +$	*fail to*	DOWN	$\equiv$	$\sqsupset$	$\sqsubset$	$\wedge$	$\smile$	$\mid$	$\#$
$- / \circ$	*refuse to*	DOWN	$\equiv$	$\sqsupset$	$\sqsubset$	$\mid$	$\#$	$\mid$	$\#$
$\circ / +$	*hesitate to*	DOWN	$\equiv$	$\sqsupset$	$\sqsubset$	$\smile$	$\smile$	$\#$	$\#$
$+ / +$	*admit that*	UP	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\wedge$	$\#$	$\#$
$- / -$	*pretend that*	UP	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\#$	$\wedge$	$\#$
$\circ / \circ$	*believe that*	NON	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$

Putting it all together

Putting it all together

Establish entailment relation between premise p **and hypothesis** h:

1. find a sequence of atomic edits $\langle e_1, \ldots, e_n \rangle$ which transforms p into h with $h = (e_n \circ \ldots \circ e_1)(p),\ x_0 = p,\ x_n = h,\ x_i = e_i(x_{i-1})$ for $i \in [1, n]$

Putting it all together

Establish entailment relation between premise p **and hypothesis** h:

1. find a sequence of atomic edits $\langle e_1, \ldots, e_n \rangle$ which transforms p into h with $h = (e_n \circ \ldots \circ e_1)(p),\ x_0 = p,\ x_n = h,\ x_i = e_i(x_{i-1})$ for $i \in [1, n]$
2. for each atomic edit e_i
    1. determine the lexical entailment relation $\beta(e_i)$ generated by e_i

Putting it all together

Establish entailment relation between premise p and hypothesis h:

1. find a sequence of atomic edits $\langle e_1, \ldots, e_n \rangle$ which transforms p into h with $h = (e_n \circ \ldots \circ e_1)(p),\ x_0 = p,\ x_n = h,\ x_i = e_i(x_{i-1})$ for $i \in [1, n]$
2. for each atomic edit e_i
    1. determine the lexical entailment relation $\beta(e_i)$ generated by e_i
    2. project $\beta(e_i)$ through the semantic composition tree of expression x_{i-1} to find $\beta(x_{i-1}, x_i)$ (atomic entailment relation for edit e_i)

Putting it all together

Establish entailment relation between premise p and hypothesis h:

1. find a sequence of atomic edits $\langle e_1, \ldots, e_n \rangle$ which transforms p into h with $h = (e_n \circ \ldots \circ e_1)(p),\ x_0 = p,\ x_n = h,\ x_i = e_i(x_{i-1})$ for $i \in [1, n]$
2. for each atomic edit e_i
    1. determine the lexical entailment relation $\beta(e_i)$ generated by e_i
    2. project $\beta(e_i)$ through the semantic composition tree of expression x_{i-1} to find $\beta(x_{i-1}, x_i)$ (atomic entailment relation for edit e_i)
3. join atomic entailment relations across the sequences of edits: $\beta(p,h) = \beta(x_0, x_n) = \beta(x_0, e_1) \bowtie \ldots \bowtie \beta(x_{i-1}) \bowtie \ldots \bowtie \beta(x_{n-1}, e_n)$

A first example

i	e_i	$x_i = e_i(x_{i-1})$	$\beta(e_i)$	$\beta(x_{i-1}, e_i)$	$\beta(x_0, x_i)$
		Stimpy is a cat.			
1	SUB(*cat, dog*)		$\mid$	$\mid$	$\mid$
		Stimpy is a dog.			
2	INS(*not*)		$\wedge$	$\hat{}$	$\sqsubset$
		Stimpy is not a dog.			
3	SUB(*dog, poodle*)		$\sqsupset$	$\sqsubset$	$\sqsubset$
		Stimpy is not a poodle.			

A first example

i	e_i	$x_i = e_i(x_{i-1})$	$\beta(e_i)$	$\beta(x_{i-1}, e_i)$	$\beta(x_0, x_i)$
		Stimpy is a cat.			
1	SUB(*cat, dog*)		$\mid$	$\mid$	$\mid$
		Stimpy is a dog.			
2	INS(*not*)		^	^	$\sqsubset$
		Stimpy is not a dog.			
3	SUB(*dog, poodle*)		$\sqsupset$	$\sqsubset$	$\sqsubset$
		Stimpy is not a poodle.			

- Result: *Stimpy is a cat* $\sqsubset$ *Stimpy is not a poodle*

A first example

i	e_i	$x_i = e_i(x_{i-1})$	$\beta(e_i)$	$\beta(x_{i-1}, e_i)$	$\beta(x_0, x_i)$
		Stimpy is a cat.			
1	SUB(*cat, dog*)		$\mid$	$\mid$	$\mid$
		Stimpy is a dog.			
2	INS(*not*)		$\wedge$	$\hat{}$	$\sqsubset$
		Stimpy is not a dog.			
3	SUB(*dog, poodle*)		$\sqsupset$	$\sqsubset$	$\sqsubset$
		Stimpy is not a poodle.			

- Result: *Stimpy is a cat* $\sqsubset$ *Stimpy is not a poodle* ✓

An example including a verb

i	e_i	$x_i = e_i(x_{i-1})$	$\beta(e_i)$	$\beta(x_{i-1}, e_i)$	$\beta(x_0, x_i)$
		We were not permitted to smoke.			
1	DEL(*permitted to*)		$\sqsupset$	$\sqsubset$	$\sqsubset$
		We did not smoke.			
2	DEL(*not*)		$\hat{}$	$\hat{}$	$\mid$
		We smoked.			
3	INS(*Cuban cigars*)		$\sqsupset$	$\sqsubset$	$\mid$
		We smoked Cuban Cigars.			

An example including a verb

i	e_i $\quad x_i = e_i(x_{i-1})$	$\beta(e_i)$	$\beta(x_{i-1}, e_i)$	$\beta(x_0, x_i)$
	We were not permitted to smoke.			
1	DEL(*permitted to*)	$\sqsupset$	$\sqsubset$	$\sqsubset$
	We did not smoke.			
2	DEL(*not*)	$\wedge$	$\wedge$	$\mid$
	We smoked.			
3	INS(*Cuban cigars*)	$\sqsupset$	$\sqsubset$	$\mid$
	We smoked Cuban Cigars.			

- Result:
  We were not permitted to smoke. | *We smoked Cuban cigars.*

An example including a verb

i	e_i	$x_i = e_i(x_{i-1})$	$\beta(e_i)$	$\beta(x_{i-1}, e_i)$	$\beta(x_0, x_i)$
		We were not permitted to smoke.			
1	DEL(*permitted to*)		$\sqsupset$	$\sqsubset$	$\sqsubset$
		We did not smoke.			
2	DEL(*not*)		$\wedge$	$\wedge$	$\mid$
		We smoked.			
3	INS(*Cuban cigars*)		$\sqsupset$	$\sqsubset$	$\mid$
		We smoked Cuban Cigars.			

- Result:
  We were not permitted to smoke. $\mid$ *We smoked Cuban cigars.* ✓

Example: De Morgan’s Laws

Example: De Morgan's Laws

De Morgan's Laws for Quantifiers

$$\neg(\forall x\ P(x)) \Leftrightarrow \exists x\ (\neg P(x))$$
$$\neg(\exists x\ P(x)) \Leftrightarrow \forall x\ (\neg P(x))$$

Example: De Morgan's Laws

De Morgan's Laws for Quantifiers

$$\neg(\forall x\ P(x)) \Leftrightarrow \exists x\ (\neg P(x))$$
$$\neg(\exists x\ P(x)) \Leftrightarrow \forall x\ (\neg P(x))$$

p	Not all birds fly.
h	Some birds do not fly.

Obviously, $p \equiv h$.

Example: De Morgan's Laws

i	e_i	$x_i = e_i(x_{i-1})$	$\beta(e_i)$	$\beta(x_{i-1}, e_i)$	$\beta(x_0, x_i)$
		Not all birds fly.			
1	DEL(*not*)		^	^	^
		All birds fly.			
2	SUB(*all, some*)		$\sqsubset$	$\sqsubset$	$\smile$
		Some birds fly.			
3	INS(*not*)		^	$\smile$	$\equiv\sqsubset\sqsupset\smile\#$
		Some birds don't fly.			

Example: De Morgan's Laws

i	e_i	$x_i = e_i(x_{i-1})$	$\beta(e_i)$	$\beta(x_{i-1}, e_i)$	$\beta(x_0, x_i)$
		Not all birds fly.			
1	DEL(*not*)		$\hat{}$	$\hat{}$	$\hat{}$
		All birds fly.			
2	SUB(*all, some*)		$\sqsubset$	$\sqsubset$	$\smile$
		Some birds fly.			
3	INS(*not*)		$\hat{}$	$\smile$	$\equiv\sqsubset\sqsupset\smile\#$
		Some birds don't fly.			

- for all 6 possible orderings of the edits, the result is the union relation $\bigcup\{\equiv, \sqsubset, \sqsupset, \smile, \#\}$
- omits only $\hat{}$ and $|$, can therefore be seen as non-exclusion relation
- not incorrect, as $\equiv$ is included, but far less informative

Putting it all together: Limitations

- join operation tends toward less informative entailment relations (union sets of relations)
- so far no knowledge about how a sequence connecting h and p can be established
- in case there are several possible sequences, which one to choose?
- no mechanism for combining information from more than one premise at a time
- lacks inference rules of classical logic, like modus ponens, modus tollens, or disjunction elimination

Conclusion

- inference method that is able to produce desired entailment relations
- covers broad variety of example problems
- not complete, leads to loss of information
- application in the *NatLog System*