Natural Logic in Natural Language Inferences

Roy Rinberg

Introduction and outline

In this paper, I seek to discuss the role of Natural Logic in computing natural language inferences. I assume a basic understanding of logic and semantic concepts, but will attempt to build from first principles so that this paper can be used as a review article for people bridging the gap between linguistics and computer science. By the end of this paper, the reader should feel familiar with Natural Logic and have a basic understanding of what research is needed in order to improve Natural Logic.

This paper draws primarily from Bill MacCartney's 2009 PhD thesis, but also relies on more modern sources like, Larry Moss, Lauri Karttunen, and Ellie Pavlick. Section 1 is structured as a lesson on the rationale behind developing Natural Logic and seeks to develop the entailment relations that will serve as the foundation for Natural Logic. Section 2 discusses the tools used for generating semantic entailments compositionally; first through lexical entailments - as in substituting 'car' for 'convertible' in the expression "red car"; and then through the projection of a semantic function onto expressions - defining how a specific edit changes the entailment relation between two x and y. Section 3 quickly puts together a generic algorithm for generating an entailment relation for any two expressions p and h. Finally, section 4 briefly discusses problems and possible research in Natural Logic.

1. What is Natural Logic and how do we arrive at it?

"Natural Logic attempts to do formal reasoning in natural language making use of syntactic structure and the semantic properties of lexical items and constructions. It contrasts with approaches that involve a translation from a natural to a formal language such as predicate calculus or a higher-order logic." (Karttunen, 2015).

For this paper, I will assume a general familiarity with monotonicity. But as a short review: Monotonicity allows us to make inferences on the entailment of subclasses and super-classes. 'With' is upwards entailing, and so, it entails more general statements. Without is downwards entailing, and so, entails more specific statements:

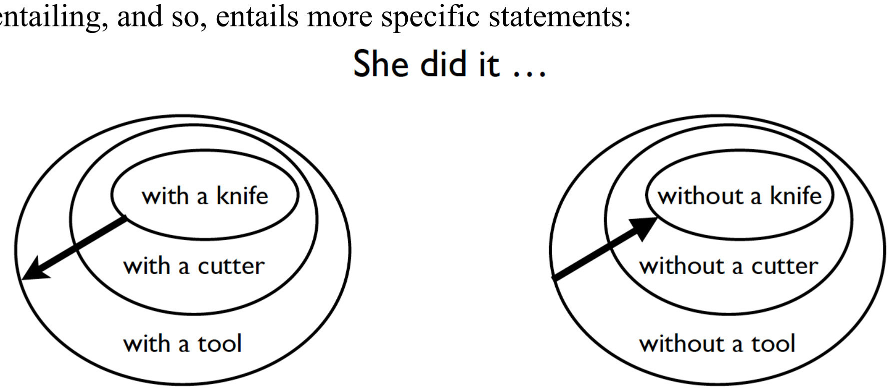


Figure 1. Image describing Monotonicity (Karttunen, 2015)

a. She did it with a knife ENTAILS she did it with a tool
b. She did it without a tool ENTAILS she did it with a knife

Monotonicity is the starting point of natural logic. But Natural Logic stems specifically from the problems generated by only looking to monotonicity. Here are two, very simple inferences humans are willing to make.

a. Nobody can enter without a valid passport $\vDash$ Nobody can enter without a passport.
b. Whiskers is a cat $\vDash$ Whiskers is not a poodle.

Monotonicity by itself provides no method for handling statement b – it lacks the ability to handle <u>semantic exclusion</u>. I wish to extend on the usefulness of monotonicity, and at the very least expand on it to include semantic exclusion. Historically, one approach semanticists have taken is to use set-theoretic categorizations for entailment, such as 3-way entailment relations:

$$\text{ENTAILMENT} \stackrel{\text{def}}{=} \{\langle p, h\rangle \in \mathbf{Dom}_T^2 : p \models h\}$$
$$\text{CONTRADICTION} \stackrel{\text{def}}{=} \{\langle p, h\rangle \in \mathbf{Dom}_T^2 : p \models \neg h\}$$
$$\text{COMPATIBILITY} \stackrel{\text{def}}{=} \{\langle p, h\rangle \in \mathbf{Dom}_T^2 : p \not\models h \land p \not\models \neg h\}$$

Where Dom_T is the domain of declarative expressions and Dom_T^2 is its Cartesian product. However, most formulations were unable to encode all types of relationship – 3-way entailment lacks reverse entailment, whereas formulations like monotonicity lack contradiction. This distinction is easily seen through the following figure:

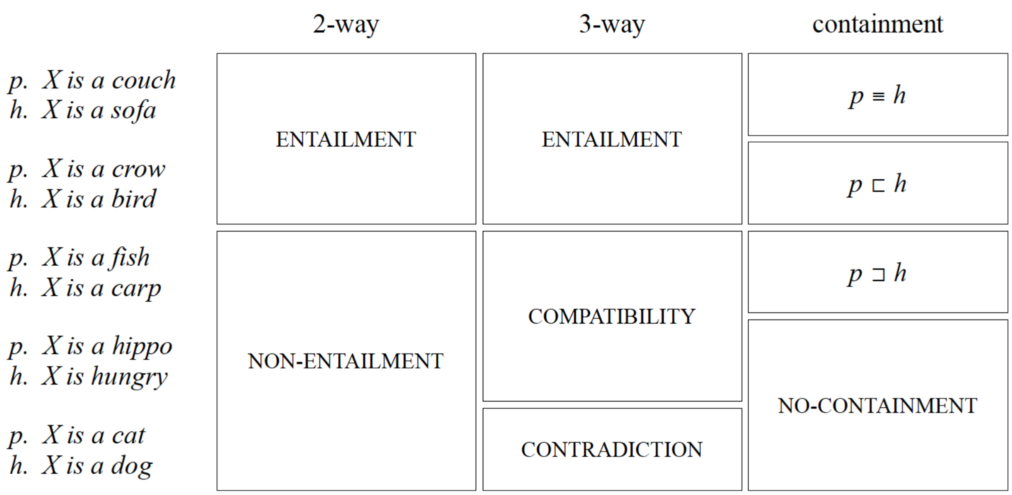


Figure 2. Image describing different entailment relations (MacCartney, 2009)

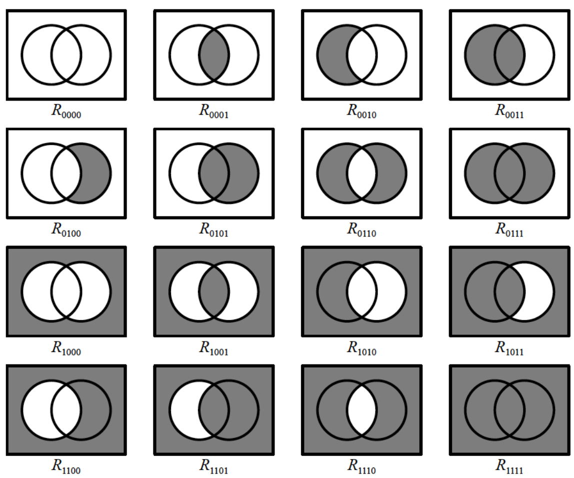
Since, for this exercise, I am designing Natural Logic from scratch, I am allowed to choose exactly what entailment relations to use. The goal here is to find a set of entailment relations that partition the space of relations between expressions, in a natural, intuitive way. Ideally, every pair of expressions can be assigned to a unique entailment relation.

It seems reasonable to accept the use of set relations to encode entailment relations, and decide on the best set of logical operations to encode the entailment relations that are important to us. Implicitly, I am now representing two expressions I wish to relate as sets of models which contain those elements that make the expressions true - though I will not use this formulation for any form of computation.

MacCartney (2009) proposes a natural set of logical operations on the two sets, generated by the Venn diagram of the two sets.

label	definition	meaning
00	$\overline{x} \cap \overline{y}$	in neither x nor y
01	$\overline{x} \cap y$	in y but not x
10	$x \cap \overline{y}$	in x but not y
11	$x \cap y$	in both x and y

Immediately, it is easy to see 16 kinds of relations between any two expressions:



relation	constraint on x	constraint on y	constraint on $\langle x, y \rangle$
R_{0000}	$\emptyset = x = U$	$\emptyset = y = U$	$x = y$
R_{0001}	$\emptyset \subset x = U$	$\emptyset \subset y = U$	$x = y$
R_{0010}	$\emptyset \subset x = U$	$\emptyset = y \subset U$	$x \supset y$
R_{0011}	$\emptyset \subset x = U$	$\emptyset \subset y \subset U$	$x \supset y$
R_{0100}	$\emptyset = x \subset U$	$\emptyset \subset y = U$	$x \subset y$
R_{0101}	$\emptyset \subset x \subset U$	$\emptyset \subset y = U$	$x \subset y$
R_{0110}	$\emptyset \subset x \subset U$	$\emptyset \subset y \subset U$	$x \cap y = \emptyset \wedge x \cup y = U$
R_{0111}	$\emptyset \subset x \subset U$	$\emptyset \subset y \subset U$	$x \cap y \neq \emptyset \wedge x \cup y = U$
R_{1000}	$\emptyset = x \subset U$	$\emptyset = y \subset U$	$x = y$
R_{1001}	$\emptyset \subset x \subset U$	$\emptyset \subset y \subset U$	$x = y$
R_{1010}	$\emptyset \subset x \subset U$	$\emptyset = y \subset U$	$x \supset y$
R_{1011}	$\emptyset \subset x \subset U$	$\emptyset \subset y \subset U$	$x \supset y$
R_{1100}	$\emptyset = x \subset U$	$\emptyset \subset y \subset U$	$x \subset y$
R_{1101}	$\emptyset \subset x \subset U$	$\emptyset \subset y \subset U$	$x \subset y$
R_{1110}	$\emptyset \subset x \subset U$	$\emptyset \subset y \subset U$	$x \cap y = \emptyset \wedge x \cup y \neq U$
R_{1111}	$\emptyset \subset x \subset U$	$\emptyset \subset y \subset U$	$x \cap y \neq \emptyset \wedge x \cup y \neq U$
			$\ldots \wedge x \not\subseteq y \wedge x \not\supseteq y$

Figure 3. 16 elementary set relations $\mathfrak{R}$ (MacCartney, 2009)

I am able to convert from set relations to entailment relations by extending the extensional meaning to an intensional meaning – namely, expressions of the same semantic type, x and y, belong to relation R_{1101} if and only if y holds in every model where x holds (but not vice-versa). This might seem problematic, as it is computational infeasible to compute an extension over an uncountable number of models; however, my usage of Natural Logic operates in the other direction – I will not provide a model and asking what entailment relation it has, but rather after assigning an entailment relation, I will ask whether or not a model has some property.

These 16 relations ($\mathfrak{R}$) encode every kind of entailment relationship between any two expressions and are the foundation for Natural Logic.

I can classify the relations, looking for relationships that human beings do not use in natural language, and easily see that nine relations in $\mathfrak{R}$, namely R_{0000}, R_{0001}, R_{0010}, R_{0011}, R_{0100}, R_{0101}, R_{1000}, R_{1010}, and R_{1100}, are boundary cases in which either x or y is either empty or universal.

Singleton degeneracies and other edge cases:

The degenerate relations: R_{0000}, R_{0001}, R_{0010}, R_{0100}, R_{1000}, R_{0011}, R_{0101}, R_{1010}, and R_{1100} cover the cases in which at least one of x or y are either empty or universal.

i.e. x =All female presidents eat pie. y = dogs who can type eat pie.

i.e. x is a female or not. y is a man or not.

However, while *contradictions and tautologies may be common in logic textbooks, they are rare in everyday speech*; these 9 degenerate relations do not tend to come up in natural language, outside of classrooms. It seems reasonable to focus on the 7 remaining relations, which are more common in natural language; MacCartney refers to these as the set $\mathfrak{B}$.

symbol[10]	name	example	set theoretic definition[11]	in $\mathfrak{R}$
$x \equiv y$	equivalence	*couch* $\equiv$ *sofa*	$x = y$	R_{1001}
$x \sqsubset y$	forward entailment	*crow* $\sqsubset$ *bird*	$x \subset y$	R_{1101}
$x \sqsupset y$	reverse entailment	*Asian* $\sqsupset$ *Thai*	$x \supset y$	R_{1011}
$x \wedge y$	negation	*able* $\wedge$ *unable*	$x \cap y = \emptyset \wedge x \cup y = U$	R_{0110}
$x \mid y$	alternation	*cat* $\mid$ *dog*	$x \cap y = \emptyset \wedge x \cup y \neq U$	R_{1110}
$x \smile y$	cover	*animal* $\smile$ *non-ape*	$x \cap y \neq \emptyset \wedge x \cup y = U$	R_{0111}
$x \# y$	independence	*hungry* $\#$ *hippo*	(all other cases)	R_{1111}

Figure 4. Seven basic entailment relations $\mathfrak{B}$ (MacCartney, 2009).

$\mathfrak{B}$ is the set that Natural Logic generally operates with, and is what I will continue to discuss in this review paper. I think it is important to note that this is a decision to make Natural Logic more convenient, not because there is a necessity to do so.

2. Compositional Semantics

2.1 Joins

At the root of semantics is the principle of compositionality – that if 'John is sad and Mary is sad', both 'John is sad' and 'Mary is sad'. To build up to this in Natural Logic, I must now generate a principle for joining entailment relations. Even on an intuitive level, this makes sense – if I know how x relates to y, and how y relates to z, I suspect I can say something about how x relates to z. MacCartney utilized the bowtie operator ⋈, Join, to generate entailments relations compositionally.

$$R \bowtie S \equiv \{< x, y >: \exists y(< x, y > \in R \wedge < y, z > \in S)\}$$

Some of these joins are possible to prove logically, specifically - joining negation. However, for the most part, rules for joins can be constructed through human intuition and putting together evidence. Some joins have very clear progressions, and generally have a clear set of governing principles.

$$\begin{aligned} \sqsubset \bowtie \sqsubset &= \sqsubset \\ \sqsupset \bowtie \sqsupset &= \sqsupset \\ \wedge \bowtie \wedge &= \equiv \\ \forall R : R \bowtie \equiv &= R \\ \forall R : \equiv \bowtie \mathrm{R} &= R \end{aligned}$$

However, it is pretty clear that not all joins will create deterministic relationships. I present MacCartney's example using alternation. With the same relationship between x and y, and y and z, I am able to generate almost every kind of relationship.

$x \mid y$	$y \mid z$	$x \; ? \; z$
gasoline $\mid$ *water*	*water* $\mid$ *petrol*	*gasoline* $\equiv$ *petrol*
pistol $\mid$ *knife*	*knife* $\mid$ *gun*	*pistol* $\sqsubset$ *gun*
dog $\mid$ *cat*	*cat* $\mid$ *terrier*	*dog* $\sqsupset$ *terrier*
rose $\mid$ *orchid*	*orchid* $\mid$ *daisy*	*rose* $\mid$ *daisy*
woman $\mid$ *frog*	*frog* $\mid$ *Eskimo*	*woman* $\#$ *Eskimo*

This problem of non-deterministic joining points to a much larger problem in computational semantics, and the overarching problem in Natural Logic – ultimately the entailment relations of words depends not on the relationships of the words, but on the semantic meaning of the words themselves. This problem will come up in other locations, but for now I will continue, wary of this issue.

2.2 Semantic composition through lexical entailments

Taking the (wrong, but simplifying) assumption: tense and aspect matter little in inferences, MacCartney (2009) claims that if two linguistic expressions differ by a single atomic edit (deletion, insertion, or substitution), then the entailment relation between them depends on two factors:

1. The lexical entailment relation generated by the edit;
$$\beta\big(x, e(x)\big) \equiv \beta(e) = X$$
2. How this lexical entailment relation is affected by semantic composition with the remainder of the expression (the context). MacCartney refers to this as 'Projectivity'
$$\beta(x, y) = Y$$
$$\beta\big(f(x), f(y)\big) = ?$$

Here β is the function mapping natural language expressions to their entailments.

2.3 Lexical entailment

The goal here is to generate some function that maps edits to entailments, such that making an edit generates an entailment relation between the original and edited sentences. There are three types of edits that are possible on a sentence: Substitution, Deletion, and Insertion.
Though, prior to proposing a set of rules, I will begin with a concrete example:

Example 1.

Let $x = red\ car$
Let $e = sub(car, convertible)$
$\beta(e) = \beta(red\ car, red\ convertible) = \sqsupset$ because convertible is a hyponym of car
$$red\ car \sqsupset red\ convertible$$

Example 2.

If I were to change the edit e
$e = del(red)$
$\beta(e) = \sqsubset$ because red is an intersective modifier
$$red\ car \sqsubset car$$

There are two important kinds of substitutions that require separate treatments: open-class terms, which contain words that fall in and out of use (most nouns, verbs, adjectives, and adverbs) and closed-class terms, which change much less frequently, and generally cannot be defined by their extension, and rather are defined by their function (words like "from", "most", and "the") (MacCartney, 2009).

2.4 Substitutions of open-class terms

Open-class term substitutions are perhaps the greatest example of the usefulness of Natural Logic. Over the past 50 years, people have created an excessive amount of information about open-class terms, and compiled them into databases (like WordNet), which encode relationships like synonymy, hyponymy, and antonymy. Conveniently, these relationships are easily aligned with corresponding entailment relations (MacCartney, 2009).

Synonyms :	≡ relation (sofa ≡couch, happy ≡ glad, forbid ≡ prohibit);
hyponym-hypernym pairs :	⊏ relation (crow ⊏ bird, frigid ⊏ cold, soar ⊏ rise);
antonyms :	| relation (hot |cold, rise | fall, advocate | opponent).

As a simple example:

a = unmarried man b = bachelor

$$\beta(sub(a, b)) = \equiv$$

2.5 Substitutions of closed-class terms

The set of closed-class terms contains within it conjunctions (and, or), articles (the, a), demonstratives (this, that), prepositions (to, from, at), and generalized quantifiers. I found the least amount of research on this, and so there is a lot of room for improvement here.

Handling demonstratives like "this" and "that" is a fairly simple process in some regimes, but can be quite difficult in Natural Logic, as there is no real mechanism for anaphora. Natural Logic, as it is now, is not used as a dynamic logic – it seeks to relate one expression to another, it does not develop relations throughout a text, building on the meaning of an expression as more information is provided. In this same point, it does not seek to account for the problem of anaphora, which is perhaps most elegantly handled by Kamp's Discourse Representation Theory (DRT); as this does not seem to be a topic currently undergoing research, a short discussion of a possible extension of Natural Logic and DRT is offered in section 4.

Substitutions of prepositions like 'from' and 'to' seem manageable, but not generalizable; it seems possible to generate a relationship between each preposition, manually, as there are a relatively small number of prepositions. I have found little work on this, and from my own reasoning it does not seem to inform us very much. The expression "I walked *to* the bank", informs us very little about if "I walked *from* the bank." For the purpose of computation, edits of prepositions should be avoided to retain as much information as possible.

Substitutions of Generalized Quantifiers, like: "Some", "All", "More than 3 elephants" are perhaps the most approachable. Most obviously, it is obvious that "every cat eats" implies that "some cats eat" (provided there are cats!). General Quantifiers are largely governed by rules of monotonicity, which are easily computed, and these lexical edits will not be made explicit here. It is important to note, however, that like all difficult linguistics question, substitutions of General Quantifiers do not end with logic, and Natural Logic is forced to engage in Gricean questions like: does "I have four children" ⊏ "I have two children", or does "I have four children" | "I have two children".

Logical conjunctions like 'and' and 'or' will be briefly discussed in the upcoming section "Semantic composition – Projectivity".

2.6 Generic deletions and insertions

The general approach to deletions and insertions is to apply monotonicity. It is clear, without much explanation that "the car which has been parked outside since last week" ⊏ "car". The original example in the Lexical Entailment section should be looked at in this lens (MacCartney, 2009).

2.7 Special cases of deletions, insertions, and substitutions

Unfortunately, generic deletions and insertions only get us so far. In the following section I will discuss two, notable cases where humans infer entailment relations that differ from what Natural Logic would predict: factives and implicatives, and non-subsective modifier adjectives. In principle, this section is the most problematic for the application of Natural Logic because it is generally difficult to tell if a phrase deserves special treatment or not.

Factives & Implicatives

Implicatives are phrases that have a specific entailment that is dependent on the polarity of the context. There exist different types of implicatives (namely, different varieties of two-way and one-way implicatives), but their premise is the same – entailment relations follow rules depending on context, which need to be accounted for. Factives and counterfactives act similarly to implicatives, except rather than entailment they encode presuppositions. This means that in all contexts, the presupposed statement is either always true or always false.

Here is a specific cases for intuition about both factives and implicatives:

Remember To: Two-way Implicative ++|--
Remember That: Factive

a. She remembered to lock the door. ENTAILS. She locked the door.
b. She did **not** remember to lock the door. ENTAILS. She did **not** lock the door.
c. She remembered that she locked the door. PRESUPPOSES. She locked the door.
d. She did **not** remember that she locked the door. PRESUPPOSES. She locked the door

There are a lot of interesting nuances in the nature of implicatives and factives, and an entire squib could be devoted to this topic alone (I suggest reviewing Lauri Karttunen's "From Natural Logic to Natural Reasoning"), however, I discuss this topic here not to do investigate proper treatment of factives and implicatives, but rather to point out that human beings often make inferences beyond the logical entailment and presupposition of implicatives and factives.

Under closer inspection, Karttunen observes that one-way implicatives, which entail a definite entailment one-way, are often are interpreted as if they were a two-way entailment, though may be 'canceled out' explicitly, which Karttunen calls **soft inferences** (2015). Here is an example of a one-way implicative "prevent" (+-)

a. The language barrier did not prevent us from sharing a few laughs.
b. Her mother did not prevent her from visiting her father, but she never did.

Both cases are perfectly acceptable to say though entail opposing inferences. Karttunen proposes that these problems are caused by conditional perfection, which pushes people to interpret simple conditionals like *if p then q* as biconditionals, *iff p then q*. In my opinion, this relates to some kind of Gricean Maxims pushing people to assume the most informative statement is being said.

Unfortunately – the process for determining if an expression behaves as a factive or implicative is not straightforward: "The sobering finding of this study that we are now in the progress of replicating with a more careful experimental design suggests that some very basic inferences such as whether the event described by an infinitival complement happened or not depend on opinions that are not part of the literal meaning of the sentence. This is a difficult problem for compositional semantics and for Natural Logic as well" – Kartunnen (2015).

Non-subsective adjectives

Touching the core of the problem with entailments and presuppositions, I now continue to Ellie Pavlick's research on non-subsective adjectives.

Here is a visual description of three, important different types of adjectives:

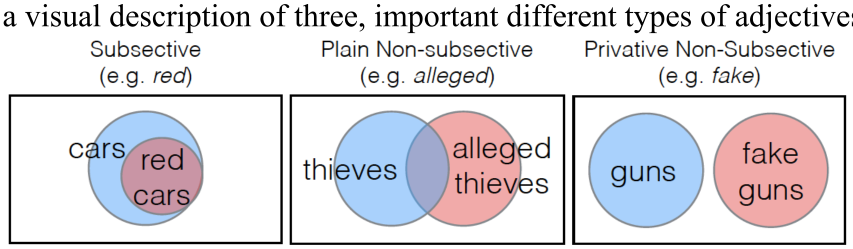


Figure 5. Categorization of Adjective types (Pavlick, 2017).

Words like fake, former, and alleged generate the | relation (a fake diamond is not a diamond), but words like alleged are less clear – an alleged thief might be a thief. Each of these types of adjectives seem to hint very heavily towards specific entailment relations

subsective adjectives:	$red\ car \sqsubset car$ monotonic	entailment
plain non-subsective adjectives:	$thief\ \#\ alleged\ thief$	unknown
privative non-subsective adjectives:	$fake\ gun\ \vert\ gun$	alternation

The question now becomes – how can one label adjectives productively? Pavlick conducted research using Amazon's Mechanical Turk program to generate empirical readings on specific words to the type and degree of entailment generated by an adjective (2017).

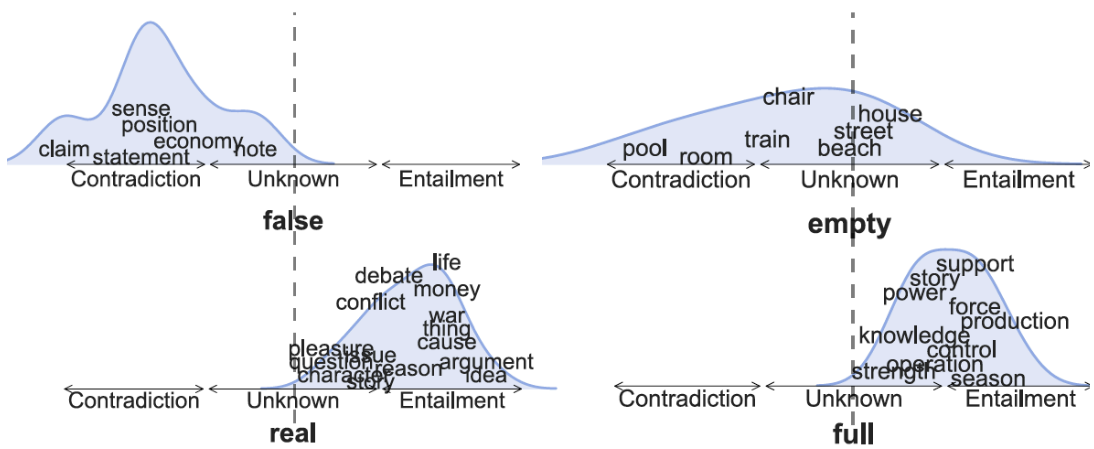


Figure 6. Illustration of entailment relations generated by insertion of an adjective. Generated by Amazon Mechanical Turk (Pavlick, 2017).

These findings unfortunately generated rather blurry delineations on the entailment of an adjective. This by itself is problematic for assign lexical entailments when adding, deleting, or substituting such an adjective, though at least provides computer scientists with a tool for generating a numeric confidence level of a type of adjective.

Pavlick goes about trying to solve the problems generated by non-subsective adjectives by claiming that nouns are assumed to be "present, salient, and relevant", and so – "modifiers that communicate presence and saliency tend to be entailed, regardless of the noun with which they are being composed or the context in which it appears, while modifiers that communicate absence or irrelevance tend to generate contradictions."

It seems that rather than being able to assign an entailment relation to a specific adjective, entailment relations depend on the word's usage in a specific sentence. To illustrate this, I present one of Pavlick's most famous examples: when asked about the sentence, "Bush travels to Michigan to remark on the economy", people respond confidently that "economy" refers to "American economy", and inserting "Japanese" before economy, would yield a contradiction. However, the sentence "Bush travels to Michigan to remark on the Japanese economy", humans agree that this entails "Bush travels to Michigan to remark on the economy" (Pavlick, 2017).

Pavlick's work provides further evidence against the usefulness of creating lexical entailment relations, arguing, similarly to Lauri Karttunen, that at the end of the day, the entailments of words depend more on the meaning of the words themselves than their relationships. Compositional semantics depends enough on the actual semantics of the words and pragmatics of the sentence that one cannot generate generalizable rules, productively.

Still, it is obvious to everyone reading this that there are problems with Natural Logic, because if it were a solved problem, no one would be worrying about natural language inferences. So, I will abstain from publishing my PhD thesis just yet, and continue to the following step in semantic composition – projection.

2.8 Semantic composition- Projections

Here I seek to address the question of how is lexical entailment relation is affected by semantic composition with respect to the remainder of the expression (the context). Namely, how does the lexical entailment *project* through an expression. Given how x and y are related, $\beta(x, y) \in \{entailment\ relations\}$, and given for any edit $f \in \{edits\}$, $\beta(x, f(x))$, can I make a claim about $\beta(f(x), f(y)) = ? \in \{entailment\ relations\}$?

To gain intuition, first I will present an example using monotonicity:
Upon hearing the sentence: *Nobody can enter without pants* – (*nobody(can((without pants) enter)*. Is it clear if 'anyone can enter without clothes?'

Pants ⊏ Clothes
Without: ↓ ('Without' is downward monotonic)
Without↓ *pants* ⊐ Without *clothes*
Can: ↑ ('Can' is upwards monotonic)
can↑ (without *pants*) enter ⊐ can (without *clothes*) enter
Nobody: ↓ ('Nobody' is downwards monotonic)
Nobody↓ (can↑ (without ↓ *pants*) enter) ⊏ Nobody(can(without *clothes*) enter)
No one can enter without clothes.

This example of monotonicity seems straightforward, akin to multiplication by +/- 1. MacCartney seeks to generalize this to any kind of entailment relation, not only semantic containment. In theory, for each f there are 7^7 (823,543) possible entailment projections signatures (in 𝔅):

≡	⊏	⊐	^	$\mid$	–	#

A	B	C	D	E	F	G

As, $\forall x \in \{A, \ldots, G\},\ x \in \mathfrak{B}$, and $|\mathfrak{B}| = 7$

This is not ideal, but it turns out that upon closer inspection, relatively few projections are really used; MacCartney's thesis maps out a few important projectivity signatures. Though these projectivity signatures cannot generally be proven rigorously, they can be shown empirically to hold in most cases. The generation of projectivity signatures is a labor-intensive job, which Amazon Mechanical Turk seems ideal for. However, for this review paper, I will accept MacCartney's signature maps as correct; and here I present a set of projectivity signatures for logical connectives:

	projectivity						
connective	≡	⊏	⊐	^	$\mid$	‿	#
negation (*not*)	≡	⊐	⊏	^	‿	$\mid$	#
conjunction (*and*) / intersection	≡	⊏	⊐	$\mid$	$\mid$	#	#
disjunction (*or*)	≡	⊏	⊐	‿	#	‿	#
conditional (*if*) (antecedent)	≡	⊐	⊏	#	#	#	#
conditional (*if*) (consequent)	≡	⊏	⊐	$\mid$	$\mid$	#	#
biconditional (*if and only if*)	≡	#	#	^	#	#	#

Figure 7. Projectivity signatures for logical connectives (MacCartney, 2009).

And for comparison, I also present the projectivity of quantifiers, which take two arguments:

	projectivity for 1st argument							projectivity for 2nd argument						
quantifier	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
some	$\equiv$	$\sqsubset$	$\sqsupset$	$\smile^\dagger$	$\#$	$\smile^\dagger$	$\#$	$\equiv$	$\sqsubset$	$\sqsupset$	$\smile^\dagger$	$\#$	$\smile^\dagger$	$\#$
no	$\equiv$	$\sqsupset$	$\sqsubset$	$\mid^\dagger$	$\#$	$\mid^\dagger$	$\#$	$\equiv$	$\sqsupset$	$\sqsubset$	$\mid^\dagger$	$\#$	$\mid^\dagger$	$\#$
every	$\equiv$	$\sqsupset$	$\sqsubset$	$\mid^\ddagger$	$\#$	$\mid^\ddagger$	$\#$	$\equiv$	$\sqsubset$	$\sqsupset$	$\mid^\dagger$	$\mid^\dagger$	$\#$	$\#$
not every	$\equiv$	$\sqsubset$	$\sqsupset$	$\smile^\ddagger$	$\#$	$\smile^\ddagger$	$\#$	$\equiv$	$\sqsupset$	$\sqsubset$	$\smile^\dagger$	$\smile^\dagger$	$\#$	$\#$
at least two	$\equiv$	$\sqsubset$	$\sqsupset$	$\#$	$\#$	$\#$	$\#$	$\equiv$	$\sqsubset$	$\sqsupset$	$\#$	$\#$	$\#$	$\#$
most	$\equiv$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\equiv$	$\sqsubset$	$\sqsupset$	$\mid$	$\mid$	$\#$	$\#$
exactly one	$\equiv$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\equiv$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$
all but one	$\equiv$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$	$\equiv$	$\#$	$\#$	$\#$	$\#$	$\#$	$\#$

Figure 8. Projectivity signatures for quantifiers connectives (MacCartney, 2009).

For intuition, I will take apart a simple example of projecting through the word "most" with a substitution. I start with the sentence "Most people were early", and ask 'how many people were late?'

Let f = most(a,b)
a = 'people'
x = 'early'
y = 'late'
$\beta(x, y) = |$
$\beta(f(a, x), f(b, y)) = |$
Most people were early | Most people were late
This same operation, with f = some(a,b)
Tells me that if 'some people were early' I do not know (#) if 'some people were late'

There are a lot of # relations and these projectivity signatures are not necessarily helpful. It does not help many scientists that 'most fish talk' # 'most birds talk' (MacCartney, 2009).

Caveats to Projectivity

For the most part these projectivity signatures are approximations (except in the case of negation, which can be proven to be exact), However, these approximations are not simply a function of statistical noise, many of them have a distinct source, which Natural Logic (as it is now) is not accounting for; the projection of a given entailment relation can depend on the value of the other argument to the function. That is, if I am given $\beta(x, y)$, and I am trying to determine its projection $\beta(f(x, z), f(y, z))$, the entailment relation can depend not only on the properties of f, but also on the properties of z. As a simple example:

Take x = French man
y = European man
z = Parisian
$\beta\left(insert(x, z), insert(y, z)\right) = \sqsubset$ as Parisian is intersective

However, in this case, it is clear that 'a Parisian, French Man' ≡ 'a Parisian, European Man'.

This caveat furthers the importance of the problem that Pavlick is addressing. The actual semantics of the words seem to matter – Natural Logic is unable to project universally.

3. Putting it all together: Algorithmically

I have now developed all the necessary prerequisites, and I now present an algorithm for creating natural language inferences using Natural Logic.

Goal: Given expressions p and h, compute some inference about h from p.
Natural Logic approach: Relate expressions p and h with an entailment relation (in $\mathfrak{B}$).

1. Find a sequence of atomic edits <e_1,…,e_n> which transforms p into h:
   h= $(e_n \circ \ldots \circ e_1) \circ p$. I define $x_i = e_i \circ x_{i-1}$
2. For each e_i
   a. Determine the lexical entailment relation $\beta(e_i) = \beta(x_{i-1}, e_i(x_{i-1}))$
   b. Project $\beta(e_i)$ to find the entailment relation $\beta(x_{i-1}, x_i)$for $i = 1, \ldots, n-1$
      For notational consistency with other work, I will write this as $\beta(x_{i-1}, e_i)$
3. Join atomic entailment relations across the sequence of edits:
   $\beta(p, h) = \beta(x_0, x_n) = \beta(x_0, e_1) \bowtie \cdots \bowtie \beta(x_{i-1}, e_i) \bowtie \cdots \bowtie \beta(x_{n-1}, e_n)$

For the benefit of the reader, it is important to state that the Manning Lab (Stanford) developed a Java program called NatLog applying this algorithm, though I will not go into specific implementations here. However, in 2009, it competed with the state-of-the-art Natural Language Inference softwares, and performed with 70.5% accuracy on the FraCas (three-way classification) test set.

System	P %	R %	Acc %
baseline: most common class	55.7	100.0	55.7
bag of words	59.7	87.2	57.4
NatLog 2007	68.9	60.8	59.6
NatLog 2008	89.3	65.7	**70.5**

Table 7.3: Performance of various systems on 183 single-premise FraCaS problems (three-way classification). The columns show precision and recall for the YES class, and accuracy.

4. Conclusions and Extensions

In this review paper, my goal has been to develop a broad understanding of Natural Logic, how it can be used, and some problems in it. I now wish to end this review with a short description of the issues currently facing Natural Logic that should be addressed, and possible directions that could be taken in addressing them.

1. The ability to infer depends heavily on non-deterministic processes of finding an appropriate edit sequence connecting p and h
2. The usefulness of the result is sometimes limited by the tendency of the join operation toward less informative entailment relations
3. People understand inferences to be <u>different</u> from what Natural Logic says they should be
4. People infer inferences <u>beyond</u> what Natural Logic tells us
5. Natural Logic is used statically

Problem 1 lends itself well to the tools found in computer science. While creative processes are difficult for computers, it is very simple to run many iterations of the same algorithm over different edit sequences, until a sufficiently good sequence is found. Probabilistic model checking algorithms and variations of randomized Monte-Carlo methods seem to be a good tool for generating appropriate edit sequences.

As for problem 2, it seems the information-lossy joining process is unavoidable; I do not have a solution, and would like to emphasize that this should not be brushed off lightly. The only solution that seems intuitive is to apply the same approach as with problem 1, and run many iterations of this algorithm, ultimately choosing the sequence of edits through which the joining process loses the least amount of information. This problem warrants serious consideration, and possibly is enough of a reason by itself to avoid Natural Logic all together.

Ultimately problem 3 and problem 4 seem to stem from the same source – a lack of world knowledge and pragmatics in computation. "If humans exploit context in order to make inferences that may not be explicitly justified by formal reasoning, our automatic systems should learn to do the same." (Pavlick, 2017). Progress in pragmatics is mandatory for progress in natural language inferences – as to how specifically this should be done. Some kind of computationally-driven approach to Gricean Maxims is a promising direction for developing more nuanced inferences. While, I will not set out to posit what such an approach might be, I believe Karttunen's consonance/dissonance effect and Pavlick's study on the explicit and implicit saliency of nouns is promising.

Problem 5 does not seem to be addressed by researchers in the field; however, I believe that accounting for this will address more than one of these problems. Natural Logic seeks to relate one expression to another, it has static representations that require precomputation, but provide no means for context-driven computation. This seems like an improper focus, as Pavlick states "The main concern from the point of view of an NLP system is not whether the set of 'imaginary cats' is a subset of the set of 'cats', but rather: can we infer that a particular mention of 'cat' is an 'imaginary cat' ". It seems reference and extension is far more important than sense and intension – and so, Natural Logic needs a means for addressing anaphora; I believe that the solution is an extension of Discourse Representation Theory (DRT), or some other dynamic logic.

DRT seeks to address the problem of anaphora by developing structures that have a mechanism for defining individuals and disambiguating referents. A possible extension of Natural Logic would be to use it in a similar format to DRT, developing Discourse Representation Structures (DRS) using Natural Logic instead of FOL, and connecting them using the techniques developed in DRT. Thus, when discussing the 'Japanese Economy' as variable x, one is able to refer to the variable x, knowing that it is indeed an 'Economy', all while retaining the usefulness of Natural Logic in generating entailments.

References:

MacCartney, W. (2009). Natural Language Inference. Stanford.

Pavlick, E. (2017. Compositional Lexical Semantics in Natural Language Inference. UPenn.

Karttunen, Lauri. 2015. From Natural Language to Natural Logic. CICLing: International Conference on Computational Linguistics and Intelligent Text Processing. pp. 295–309

Moss, L (Unpublished). Logic from Language. Indiana University, Bloomington

van Eijck, Jan. 2005. Discourse Representation Theory. Plato Stanford Encyclopedia.