A Tableau Prover for Natural Logic and Language

Long paper at EMNLP-15
Lisbon, 17–21 September

Lasha Abzianidze
TiLPS, Tilburg University, the Netherlands

Supported by NWO

Introduction

Natural language inference is one of the generic problems of natural language understanding. To account for this problem, I develop a **theorem prover** based on an analytic **tableau method** for **Natural Logic**, a logic terms of which resemble linguistic forms. The prover employs syntactically and semantically motivated schematic rules that makes its decision procedure transparent. Unlike most of the **RTE** systems, it can also reason over **several premises**. Pairing the prover with a syntactic parser and a preprocessor generating logical forms results in a proof system for natural language. After evaluation on the unseen **SICK data**, the system obtains **competitive accuracy** (81%) with almost **perfect precision** (98%).

Natural Tableau

Natural Tableau is an analytic tableau system for Natural Logic [7]: a proof system for a version of **high-order logic** based on a simply typed **λ calculus** with lexical constants. The formulas of the logic are called **Lambda Logical Forms (LLFs)** and resemble linguistic forms:

$\texttt{some}_{(et)(et)t}\,\texttt{bird}_{et}\,(\texttt{not}_{(et)et}\,\texttt{fly}_{et}) \rightarrow \texttt{not}_{((et)(et)t)(et)(et)t}\,\texttt{all}_{(et)(et)t}\,\texttt{bird}_{et}\,\texttt{fly}_{et}$

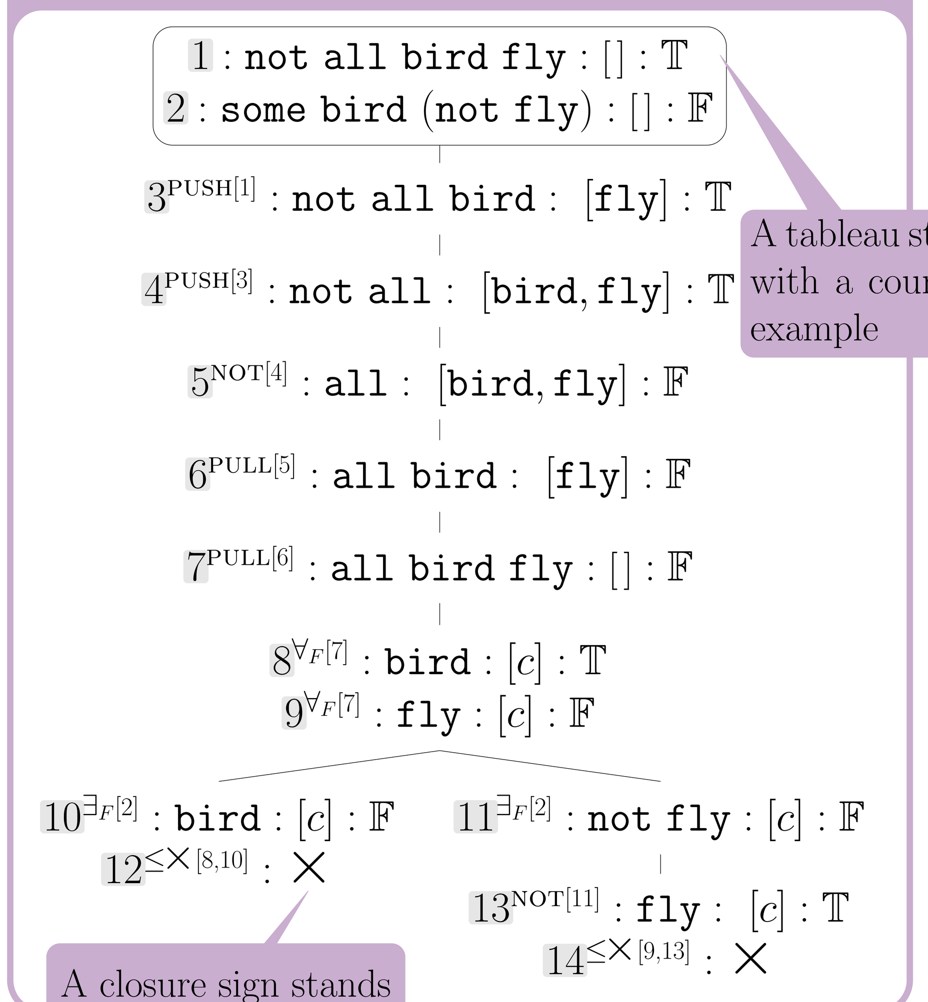
Tableau rules

$$\frac{A\,B : [\vec{C}] : \mathbb{X}}{A : [B, \vec{C}] : \mathbb{X}}\text{PUSH} \qquad \frac{A : [B, \vec{C}] : \mathbb{X}}{A\,B : [\vec{C}] : \mathbb{X}}\text{PULL}$$

$$\frac{\texttt{all}\ A\ B : [\,] : \mathbb{F}}{A : [c] : \mathbb{T} \quad B : [c] : \mathbb{F}}\forall_F \qquad \frac{\texttt{some}\ A\ B : [\,] : \mathbb{F}}{A : [d] : \mathbb{F} \mid B : [d] : \mathbb{F}}\exists_F$$

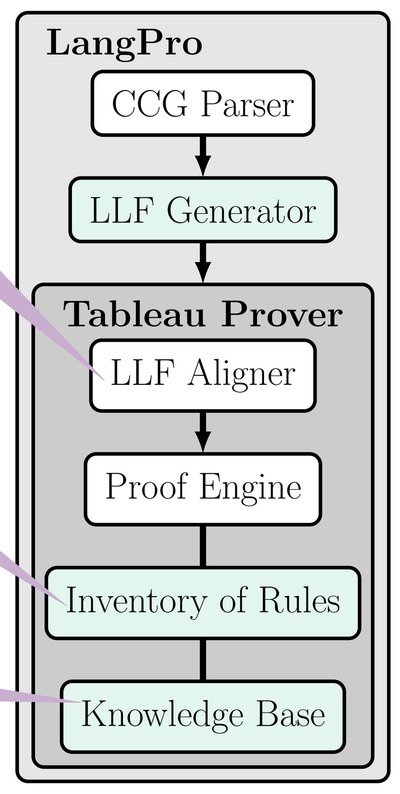
$$\frac{\texttt{not}\ A : [\vec{C}] : \mathbb{X}}{A : [\vec{C}] : \bar{\mathbb{X}}}\text{NOT} \qquad \frac{A : [\vec{C}] : \mathbb{T} \quad B : [\vec{C}] : \mathbb{F}}{\times}\leq\!\times,\ A \leq B$$

A tableau tries to refute an argument



LangPro: architecture & algorithm

Chaining a CCG parser, the LLG generator and the Natural Tableau prover results in a theorem prover for natural language, LangPro.




```
input: ({p_i}^n_1, h);          //accepts multiple premises
if  CCGparse(p_i, P_i)
    & CCGparse(h, H)
    & LLFgen(P_i, [𝒫_i, ...])
    & LLFgen(H, [ℋ, ...])
then
    LLF_Aligner([𝒫_1, ..., 𝒫_n, ℋ]);optional
    case  tab{𝒫_i:𝕋, ℋ:𝔽}, tab{𝒫_i:𝕋, ℋ:𝕋}
       CLOSED, OPEN:    return ENTAILMENT;
       OPEN, CLOSED:    return CONTRADICTION;
       OPEN, OPEN:      return NEUTRAL;
       CLOSED, CLOSED:  return ENTAILMENT;
                        report Non-determinism;
else
    return NEUTRAL;
    report Obtaining LLFs failed;
```

Online demo: `http://tinyurl.com/emnlp-langpro`

Producing LLFs for wide-coverage text

Since a function-argument application is central for both, LLFs and Categorial Grammars, it is reasonable to obtain LLFs from **Combinatory Categorial Grammar (CCG)** [8] derivations produced by the state-of-the-art CCG parsers—the **C&C parser** [2, 3] and **EasyCCG** [4].

LLFs can serve as an abstract logical form

Producing an LLF from a CCG derivation tree consists of several steps:

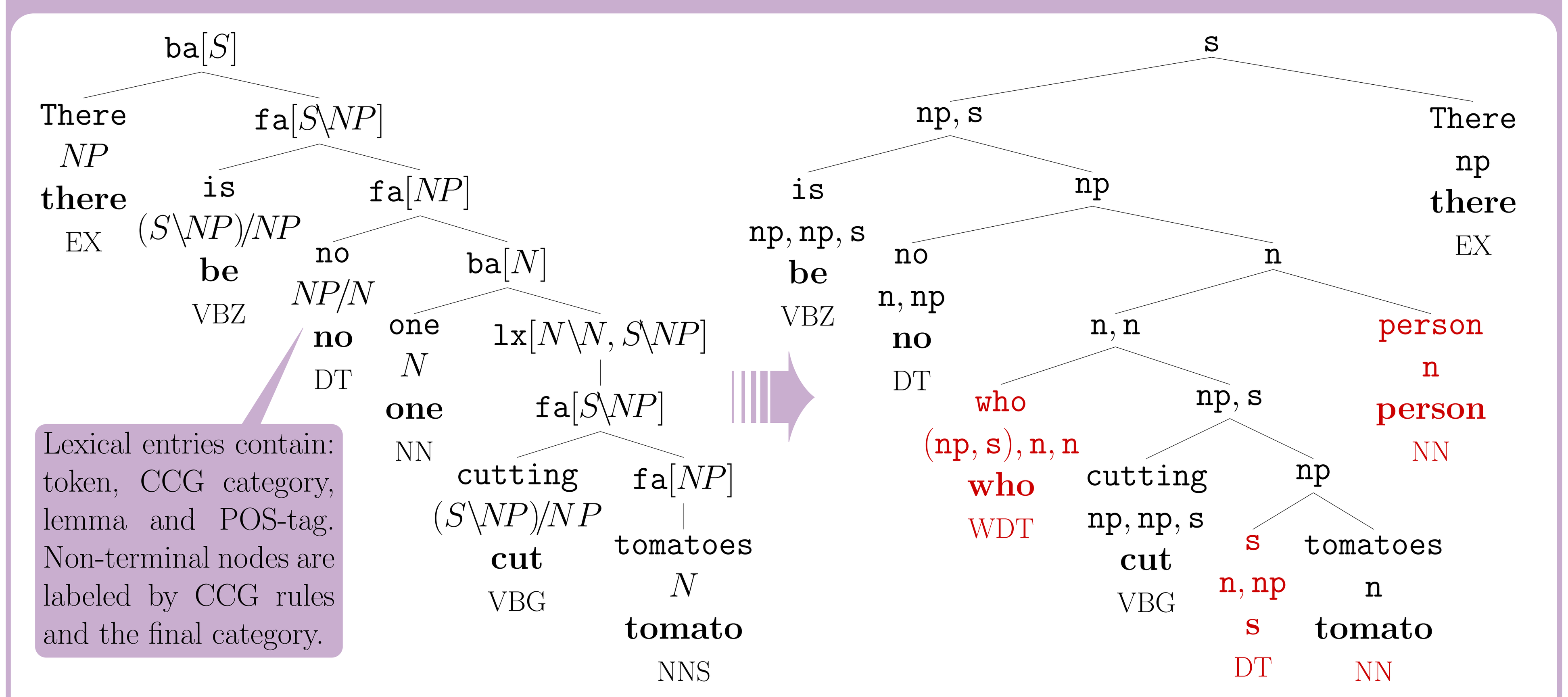
CCG Tree → CCG Term → Fixed CCG Term → LLFs (→ FOL, → DRT)

Removing directionality:
$Y\backslash X$ and $Y/X \rightsquigarrow (x, y)$
$\texttt{ba}(A_X, F_{Y\backslash X}) \rightsquigarrow F\,A$
$\texttt{fxc}(F_{Z/Y}, A_{Y\backslash X}) \rightsquigarrow \lambda x.\,F(Ax)$
$\texttt{tr}(T/(T\backslash X), A_X) \rightsquigarrow \lambda x.\,x A$
$\texttt{lx}(Y, A_X) \rightsquigarrow [A_X]_Y$
$\texttt{conj}(C_{conj}, A_X) \rightsquigarrow C_{X,X,X} A$

Correcting CCG terms:
$[\texttt{Dow}^{PER}_{n,n}\ \texttt{Jones}^{PER}_{n}]_{np} \rightsquigarrow \texttt{Dow\_Jones}_{np}$
$\texttt{nobody}_{np} \rightsquigarrow \texttt{no}_{n,np}\ \texttt{person}_n$
$[\texttt{ice}_n]_{np} \rightsquigarrow \texttt{a}_{n,np}\ \texttt{ice}_n$
$[\texttt{two}_{n,n}\ \texttt{dogs}_n]_{np} \rightsquigarrow \texttt{two}_{n,np}\ \texttt{dogs}_n$
$[\texttt{working}_{np,s}]_{n,n} \rightsquigarrow \texttt{who}_{(np,s),n,n}\ \texttt{working}$
$\texttt{who}\ V(Q_{n,np}N) \rightsquigarrow Q_{n,np}(\texttt{who}'\,V\,N)$

Type-raising quantified NPs. In case of several quantified NPs, due to scope ambiguity, several LLFs are obtained.

Example of converting a C&C CCG tree into a fixed CCG term and then LLFs



There is no one cutting tomatoes ⇝ **be(no(who(cut(s tomato))person))there**

The LLFs obtained from the fixed CCG term:

no person with a wide scope: **no (who ($\lambda x.$ s tomato ($\lambda y.$ cut $y\,x$)) person) ($\lambda z.$ be z there)**

tomatoes with a wide scope: **s tomato ($\lambda y.$ no (who (cut y) person) ($\lambda z.$ be z there))**

Conservative extension of a type system = syntactic types + semantic types

✘ LLFs only with basic semantic types e, t: poor context matching and poor NL generation;

✔ LLFs with both syntactic and semantic types: syntactic information enriches context matching (important for tableau rule applications) and facilitates generation of hypotheses.

A **sub-typing** relation over types allows interaction between syntactic and semantic types:

(a) $e \sqsubseteq \texttt{np}$, $\quad \texttt{s} \sqsubseteq t$, $\quad \texttt{n} \sqsubseteq et$, $\quad \texttt{pp} \sqsubseteq et$;

(b) $(\alpha_1, \alpha_2) \sqsubseteq (\beta_1, \beta_2)$ iff $\beta_1 \sqsubseteq \alpha_1$ and $\alpha_2 \sqsubseteq \beta_2$;

Hence, LLFs like $\textbf{tomato}_n\ c_e$ and $\textbf{on}_{np,pp}\ c_e\ d_e$, found in tableaux, are well-formed LLFs of type t. Here, an extra typing rule is used that says: if A is of type α and $\alpha \sqsubseteq \beta$, then A is of type β too.

Learning on SICK-trial / Development on SICK-train / Evaluation on SICK-test

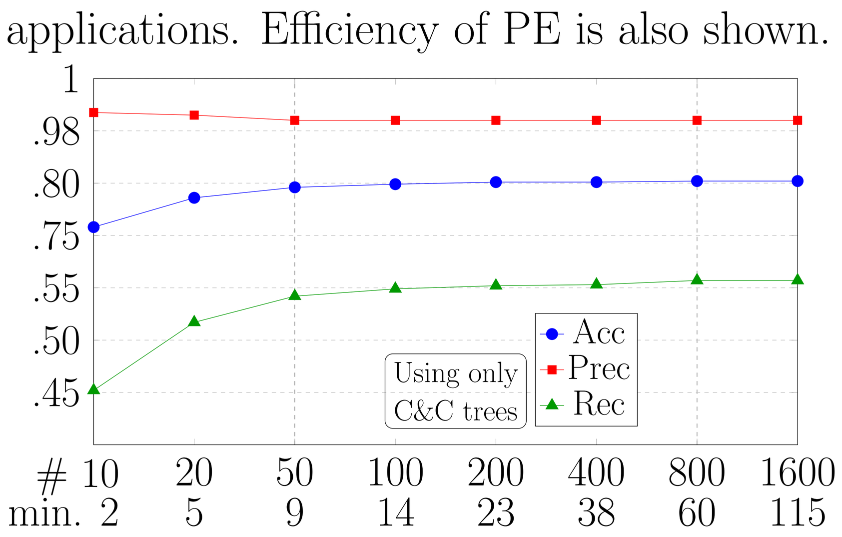
Learning on SICK-trial (500 problems) [5]:

Learning procedure

```
input: ({P_i}^n_1, H, answer);          //C&C parse trees
1  LLFgen(P_i, [𝒫_i, ...]); LLFgen(H, [ℋ, ...]);
2  case answer, tab{𝒫_i:𝕋, ℋ:𝔽}, tab{𝒫_i:𝕋, ℋ:𝕋}
      ENTAILMENT,  CLOSED,  OPEN:    halt;
      CONTRADICT,  OPEN,    CLOSED:  halt;
      NEUTRAL,     OPEN,    OPEN:    halt;
3  otherwise
4*   if 𝒫_i or ℋ is incorrect then
         try to fix LLFgen; go to 1;
5*   else if a rule or a relation is missing then
         add it to KB or IR, resp.; go to 2;
6    else halt;
```

* Carried out manually

Development defines an **efficient** (50) and an **effective** (800) upper bounds for rule applications. Efficiency of PE is also shown.



The number of rule applications (#) & runtime for 2-core 2.4GHz CPU

Error Analysis on SICK-trial & train: false positives represent dubious cases; parser errors are often a reason for failure.

Gold/LP	Problem: (premise, conclusion)
E/N	It is raining on a walking man A man is walking in the rain
E/N	There is no girl in white dancing A girl in white is dancing
N/E	Someone is playing with a toad Someone is playing with a frog
C/C	A man is playing a guitar A man is not playing a guitar
N/C	A couple is not looking at a map A couple is looking at a map

On SICK-test LangPro gets high results and proves problems solved by none of top systems:

The woman is not wearing glasses or a headdress → ¬ A woman is wearing an Egyptian headdress

Systems at SemEval [6]	Prec%	Rec%	Acc%
Illinois-LH	81.56	**81.87**	**84.57**
ECNU	84.37	74.37	83.64
UNAL-NLP	81.99	76.80	83.05
SemantiKLUE	85.40	69.63	82.32
The Meaning Factory	93.63	60.64	81.59
LangPro Hybrid-800	**97.95**	58.11	81.35
*NutCracker	-	-	78.40
Baseline (majority)			56.69

Combines answers based on both C&C and EasyCCG trees

C&C-based: 79.93
EasyCCG-bsd: 79.05

Conclusion

I extended Natural Tableau [7] in terms of rules and a formal language of LLFs. Based on it, a wide-coverage theorem prover was implemented and evaluated.

✔ LangPro is a modular system with an **explanatory decision procedure**;

✘ it heavily hinges on CCG parsing and the **learning is expensive**;

✔ it shows competitive results on SICK with almost **perfect precision**;

✔ it solves multi-premised problems with **Boolean** and **monotonic** expressions.

Also reliable due to rule-based nature

References

[1] Lasha Abzianidze. 2015. Towards a wide-coverage tableau method for natural logic. In T. Murata et al. eds, New Frontiers in Artificial Intelligence, vol 9067 of LNAI, pp 66–82. Springer.

[2] Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. CL, 33.

[3] Matthew Honnibal, James R. Curran, and Johan Bos. 2010. Rebanking ccgbank for improved np interpretation. ACL, pp 207–215.

[4] Mike Lewis and Mark Steedman. 2014. A* ccg parsing with a supertag-factored model. In Proceedings of EMNLP, pp 990–1000.

[5] Marco Marelli et al. 2014. A sick cure for the evaluation of compositional distributional semantic models. In Proceedings of LREC.

[6] Marco Marelli et al. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. SemEval 2014, pp 1–8.

[7] Reinhard Muskens. 2010. An analytic tableau system for natural logic. In M. Aloni, H. Bastiaanse, T. de Jager, and K. Schulz, eds, Logic, Language and Meaning, vol 6042 of LNCS, pp 104–113. Springer.

[8] Mark Steedman. 2000. The Syntactic Process. MIT Press.