First Experiments with Data Driven Conjecturing

Karel Chvalovský, Thibault Gauthier, and Josef Urban

CIIRC CTU

Conjecturing — some bits from history

There have been various attempts, e.g.,

- Wang in the late 1950's
  - novel and "interesting" mathematical statements
- Lenat's AM (Automated Mathematician)
- Fajtlowicz's Graffiti in the late 1980's
  - graph theory, number theory, chemistry
  - some conjectures proved by humans and published
- HR
  - number theory
- Theorema
  - algebra
- Daikon
  - invariant detector

Usually hand-crafted and/or very domain specific heuristics.
Morever, they rarely scale beyond toy examples.

What are we aiming for?

- we really do not want to produce (not yet)
  - deep and hard conjectures interesting for humans, or
  - cut formulae that make our proofs significantly shorter
- our goal here is modest — to produce some new simple variants of already known statements (analogies)

Our problem

Input

```
x <= y & x is positive implies y is positive
```

```
! [ B1 : v1_xreal_0 ] : ! [ B2 : v1_xreal_0 ] : ( ( r1_xxreal_0 ( B1 , B2 ) &
sort ( B1 , v2_xxreal_0 ) ) => ( sort ( B2 , v2_xxreal_0 ) ) )
```

Output

```
x >= y & x is negative implies y is negative
x <= y & y is negative implies x is negative
x >  y & x is negative implies y is negative
```

Word embeddings

In NLP word embeddings have proven to be very successful. A word is represented by a low dimensional vector of real numbers. The aim is to capture the meaning of words.

Properties

- cosine similarity—the similarity of two words correlates with the cosine of the angle between their vectors
- analogies

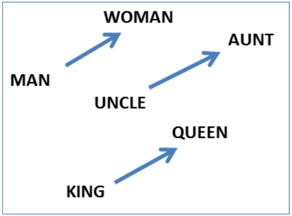


image: Mikolov et al. 2013

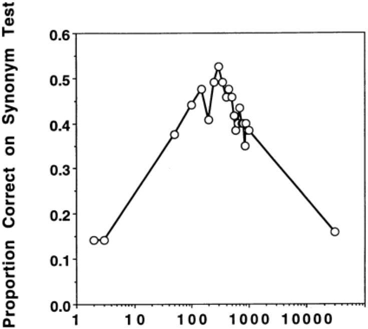
How do we obtain word embeddings?

Various approaches, but usually we use unsupervised learning and exploit the distributional (Firth's) hypothesis:

> You shall know a word by the company it keeps! [Firth 1957]

Low dimensional vectors

The quality is improved by compression. For example, the low dimension of vectors improves their semantic properties.



[Landauer and Dumais 1997]

Differences

Although the language of mathematics is a fragment of natural language, they differ significantly in many ways, e.g., in (formal) mathematics we have

- parse trees for free
- variables (they can represent any possible term) and are of unlimited supply
- a very complicated internal structure of terms (and formulae) and this structure really matters
  - we have long dependencies
  - order of tokens is important
  - a change of notation can lead to different results, e.g., a prefix notation

Representations of formulae

- there have been various attempts, e.g.,

  Sperduti, Starita, and Goller: Learning Distributed Representations for the Classification of Terms, IJCAI 1995

  - they take advantage of the tree structure of terms
- we attempt to do something similar without using the tree structure of formulae, but sometimes "sub-word" information (fasttext) is taken into account

```
! [ B1 : v1_xreal_0 ] : ! [ B2 : v1_xreal_0 ] : ( ( r1_xxreal_0 ( B1 , B2 ) &
sort ( B1 , v2_xxreal_0 ) ) => ( sort ( B2 , v2_xxreal_0 ) ) )
```

- note that it is known that using directly, e.g., word2vec, for deciding whether a propositinal formula is a tautology leads to poor results

Analogies

Say we want to produce

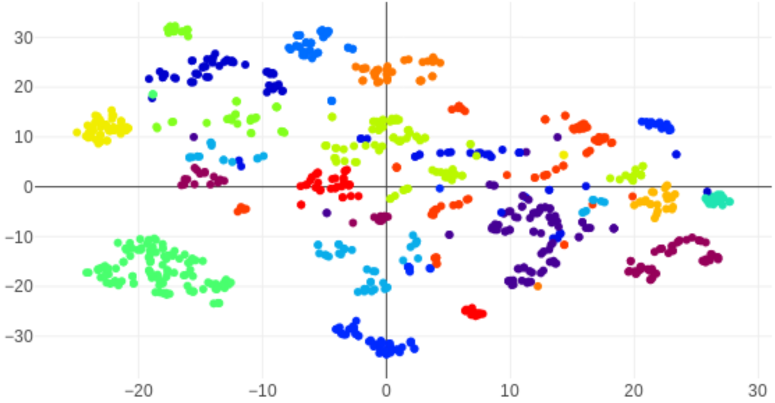
`x >= y & x is negative implies y is negative`

from

`x <= y & x is positive implies y is positive`

- we can extract the most important notions from the statement using a variant of tf-idf, see Arora et al. 2017, and shift them
  - `positive` $\rightsquigarrow$ `negative`, `<=` $\rightsquigarrow$ `>=`
- this can also be used to produce the embeddings of statements from embeddings for tokens

Representations of formulae in Mizar articles (after disambiguation)



t-SNE

Does it work?

- it is quite safe to say NO, because it produces poor results
- however, for conjecturing we do not need perfect matches, we can do some k-NN and use it for pruning the space of all possibilities
- it probably suffers from a relatively small dataset (57K statements), but results are not improving much if we take also whole proofs into account
- another drawback is that all the shifts have to play together nicely and it is hard to achieve that, moreover, there is already a way how to partially overcome this
- arguably, the main issue is that the model is too simple even for our purposes

Conjecturing as a translation task

- the task is to

  translate a statement s into a conjecture t

- we can train it as a supervised task where we have for a statement s many statements $t_1, \ldots, t_n$ that are somehow relevant to s and hence we have training pairs

$$(s, t_1), (s, t_2), \ldots, (s, t_n)$$

- we already have a list of valid statements and we can produce pairs of relevant statements from them in many ways

First experiments

- a simple example is that we can say that two statements are relevant if they share a common abstract pattern, e.g., commutativity, associativity
- we obtain 16K patterns using Gauthier's patternizer that generalize at least two statements
- they give us 1.3M (non-unique) translation pairs for NMT (with attention)
- from 30K unique formulae (statements) on the test set we get 16K new formulae (not in MML)
- 8839 of them are correct FOF formulae (660 trivial tautologies)
- using 128 most relevant premises we get
  - 5745 disprovable formulae (mainly using Paradox)
  - 1447 provable formulae
  - 987 formulae with unknown status

A simple example

We obtained

$$(X \cap Y) \setminus Z = (X \setminus Z) \cap (Y \setminus Z)$$

from

$$(X \cup Y) \setminus Z = (X \setminus Z) \cup (Y \setminus Z).$$

Examples of false but syntactically consistent conjectures

```
for n, m being natural numbers holds
n gcd m = n div m;
```

```
for R being Relation holds
with_suprema(R) <=> with_suprema(inverse_relation(R));
```

Possible future directions

- use type-checking and tree structures
- attention gives us the importance of tokens for free
- modify beam search
- many possible definitions of relevant statements, e.g., they have close representations
- many possible translation tasks, e.g., translate a statement about sets into a statement about lattices, or use a seed
- increase the training set by adding new translations
- unsupervised tasks, e.g., we have different formal libraries and we can connect them through shared notions
- however, we should also say what is a good conjecture

> *... a mathematical idea is "significant" if it can be connected in a natural and illuminating way with a large complex of other mathematical ideas.*
>
> G. H. Hardy

Thank you!