Dictionary-based learning of conditionals

Aleksandra Samonek
CEFISES, Université catholique de Louvain
aleksandra.samonek@uclouvain.be

Abstract

The goal of my doctoral projects it to propose a formal interpretation of conditional sentences which allows explaining the difference in evaluating "twin" sentences like the following:

(1.) `If Oswald didn't kill Kennedy, Kennedy is alive today.`
(2.) `If Oswald hadn't killed Kennedy, Kennedy would be alive today.`

The interpretation offered here is multi-layered. Namely, we start with a base *quasi*-ontology expressed in terms of dictionaries, understood as in computer science. Then probabilities are assigned to various keys in the dictionaries based on the connections between concepts, which the dictionaries represent. Knowledge updates are performed on various keys in the dictionaries. In order to consider a counterfactual scenario, we create a copy of an existing dictionary with the alteration of a particular key.

Dictionary construction

In order to obtain a dictionary and a network of dictionaries, we take the following steps.

Step 1. Mining the natural language. From the sentences of *the natural language* (NL), we obtain a "translation" into `simplified language` (SL). For example, from the NL-sentence of the form

Oswald killed Kennedy.

we obtain the following sentences of the simplified language:

(1.) `Oswald has killer(Kennedy)`
(2.) `Kennedy has killed_by(Oswald)`
(3.) `killing(killer, killed_by) has (Oswald, Kennedy)`

The above SL-sentences are spelled out as (*ad.* (1.)) the concept of Oswald has in it a property of being a killer (of Kennedy); (*ad.* (2.)) the concept of Kennedy has in it a property of being killed_by (Oswald)(*ad.* (3.)) the concept of killing is in it the concept of Oswald as (that who is a) killer and Kennedy as (that who is) killed_by.

Caveat: for the moment we are only concerned with relatively simple NL-sentences, which concern objects and the relations between them. Thus, certain elements of NL, *e. g.* some adverbial phrases, will be omitted.

How to extract SL-sentences from NL-sentences?
A variety of powerful extraction techniques is available in the field of machine learning (ML) applications. ML-text processing involves three phases: (i.) data preprocessing, (ii.) feature extraction, and (iii.) ML-algorithm fitting ([1, 7, 15]). Data preprocessing involves (a.) removing irrelevant punctuation and tags, (b.) removing predefined stop words, like *the*, *a* and others, (c.) tokenization, that is converting sentences into words, (d.) stemming, that is reducing words to a root, or (e.) lemmatization, an alternative to stemming which involves determining the part of speech and using the pre-existing database of the language. With the use of such methods, extracting SL-sentences from NL-sentences is attainable.

Step 2. Creating dictionaries & the semantic network
In the SL-sentences, we interpret the word `has` as *having inside as a key* and define dictionaries in the following way:

```
Oswald { killer(Kennedy): ___ , }
Kennedy { killed_by(Oswald): ___ , }
killing { (Oswald, Kennedy): ___ , }
killer {Oswald(Kennedy): ___ , }
killed_by {Kennedy(Oswald): ___ , }
```

- As concepts define all elements of SL-sentences other than the word `has`.
- Each concept gets its own dictionary, composed of ordered pairs of the form $\langle$ key : value$\rangle$, where key is the name of any dictionary (or a collection of dictionaries) and value (above marked as `___`) is any $r \in \mathbb{R}$ such that $r \in [0, 1]$.
- Certain dictionaries take other pre-defined dictionaries as arguments. We acknowledge this fact by writing:

```
killing(killer, killed_by) { (Oswald, Kennedy): ___ , }
killer {Oswald(Kennedy): ___ , }
killed_by {Kennedy(Oswald): ___ , }
```

  In what follows the value of each key in `killing` is a function of the values of argument-keys. This step allows for a multi-layer value extraction later on. Call such dictionaries **rank-2 dictionaries**.
- If one concept is a key in the dictionary of another, call the concepts **connected**. By taking concept dictionaries as vertices and connections between them as nodes in a graph, we obtain a network which represents our concept semantics. For example:

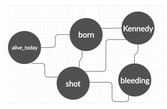

Figure 1: Concepts network example.

Open question. Could we define the strength of connections between the concept dictionaries in a non-binary and non-trivial way? How would this influence the value assignment?

Updating on dictionaries

Probability assignment. Consider a dictionary $dict = \{key_i : r_i\}$, where key_i represents the keys $(i = 1, \ldots, n)$ and r_i represents values assigned to those keys such that each value $r \in [0, 1]$. Take r to be the conditional probability of key given $dict$.
What does conditional probability mean for dictionaries? A more familiar concept of conditional probability is that for propositions: $P(A|B)$, where A and B are propositions representing events (a proposition M is true iff an event $\mathbb{M}$ occurs). Conditional probability is defined in a standard way, that is, for any events $\mathbb{A}, \mathbb{B}$:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

Fortunately, the dictionary-key relationship allows for the following translation into propositions. Suppose `red` is in the dictionary of `hot` with the value of 0.34. We take it to mean: with the probability of 0.34, if x is hot, then x is red. In other words, the conditional probability of x being red (R) given that x is hot (H) is 0.34, $P(R|H) = 0.34$.

- **Rule. If the value of a given key in a dictionary is 0.5, remove this key from the dictionary.** (This rule removes irrelevant connections from the network.)
- This form of updating is related to both deterministic action models, as in [2, 4, 5] and to Bayesian learning for neural networks (*cf.* [6, 17, 12]).
- Underlying the idea of calculating the values for keys in 2-rank dictionaries is that of **algorithmic theory of meaning** (*cf.* [10, 8, 16]).
- For now, we allow that the values of keys in dictionaries is calculated with any empirically testable method of assigning probabilities, *e. g.* as in sentiment analysis ([13, 9, 11]).

Counterfactual Embeddings

To consider a counterfactual case, take a dictionary $dict$ and create a copy $dict^*$ which differs from $dict$ in only one key-value (if the value becomes 0.5, the key is removed). Then replace $dict$ with $dict^*$ in the network and update the set of vertices V^k connected to $dict^*$, then update V_{k+1} connected to V^k, and so on.

- A copy and its generator can co-habitate the same network. In human psychology, this task corresponds to significant energetic penalty in humans (*cf.* [3, 14]).
- The operation of replacing a dictionary with its copy corresponds to the basic intuition underlying the Ramsey Test.
- Subsequent updates allows for very subtle update tracking. We obtain specific information on *why* the change in probabilistic value occurred for a given key in a given dictionary.

Reconstructing inference conditions: example

Suppose we want to define relevant entailment on the above network and suppose the propositions p and q are of the following form:
p is `A has B` and
q is `A has C`.

Relevant entailment. We say that p relevantly implies q $(p \supset q)$ iff:
(1.) `B` and `C` are connected (in the sense defined above), and
(2.) `B` makes `C` probable, that is the value of `C` in `A` is above the minimum threshold of being probable (for example $P(C|B > 0.7$, and
(3.) the value of `C` falls in the absence of `B`.

Advantages

The proposed approach:

- provides an alternative to the standard interpretation of propositions in natural language sentences in the spirit of taking each proposition to express a property of an object,
- yields a similar interpretation of conditional sentences as the possible-world approaches, but there is no talk of possible worlds as objects: in case of counterfactual conditionals we will create a copy of a dictionary and tentatively tweak probabilities in it, no other objects are postulated,
- creates a bridge (via the same underlying ontology) between the procedure for a learning function and logical inferences,
- allows testing on neural networks (one can limit the number of concepts being used and generate sentences in order to test whether the neural network will yield results similar to the learning function),
- allows the learner to spontaneously generate correct conditional sentences based on the probability ordering within the dictionary (by picking the key with the highest probability and creating a subsentence with this key).

References

[1] Charu C. Aggarwal and Cheng Xiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[2] Eyal Amir and Allen Chang. Learning partially observable deterministic action models. *Journal of Artificial Intelligence Research*, 33:349–402, 2008.

[3] Sarah R Beck, Elizabeth J Robinson, Daniel J Carroll, and Ian A Apperly. Children's thinking about counterfactuals and future hypotheticals as possibilities. *Child Development*, 77(2):413–426, 2006.

[4] Thomas Bolander and Nina Gierasimczuk. Learning actions models: Qualitative approach. In *International Workshop on Logic, Rationality and Interaction*, pages 40–52. Springer, 2015.

[5] Thomas Bolander and Nina Gierasimczuk. Learning to act: qualitative learning of deterministic action models. *Journal of Logic and Computation*, 28(2):337–365, 2017.

[6] Zoubin Ghahramani. Learning dynamic bayesian networks. In *International School on Neural Networks, Initiated by IIASS and EMFCSC*, pages 168–197. Springer, 1997.

[7] Vasileios Hatzlvassiloglou, Judith L. Klavans, and Eleazar Eskin. Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning. In *1999 Joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, 1999.

[8] Eleni Kalyvianaki. Factual content in algorithmic natural language semantics. *Proceedings of the Twelfth ESSLLI*, pages 123–133, 2007.

[9] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[10] Yiannis N Moschovakis. A logical calculus of meaning and synonymy. *Linguistics and Philosophy*, 29(1):27–89, 2006.

[11] Ramanathan Narayanan, Bing Liu, and Alok Choudhary. Sentiment analysis of conditional sentences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 180–189. Association for Computational Linguistics, 2009.

[12] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[13] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1–2):1–135, 2008.

[14] Elizabeth J Robinson and Sarah Beck. What is difficult about counterfactual reasoning. *Children's reasoning and the mind*, pages 101–119, 2000.

[15] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[16] Jakub Szymanik. Computational complexity of polyadic lifts of generalized quantifiers in natural language. *Linguistics and Philosophy*, 33(3):215–250, 2010.

[17] Eric A Wan. Neural network classification: A bayesian interpretation. *IEEE Transactions on Neural Networks*, 1(4):303–305, 1990.