Introduction to Semantic Role Labelling

Behrang QasemiZadeh
zadeh@phil.hhu.de
Computational Linguistics Department, HHU – DRAFT

October 2018–January 2019

Classification: A Gentle Introduction

We look at the general idea behind categorization as an inference mechanism and attach it as an *additional introduction* to the start of Chapter 3 of Palmer et al. (2010).

Classification: Introduction

Classification, also called Categorization, refers to a paradigm of *thinking*.

Thinking is a topic in Philosophy and what comes underneath it (look at our university's hierarchy of department and how they are *categorized*).

Here, we consider Classification as an inference mechanism for automated problem solving, in particular, for the purpose of semantic labeling.

Classification: Introduction: Inference

Inference is the process of creating a new piece of information based on (known and/or unknown) information.

> Let me clarify what I mean by known and/or unknown: The *unknown* information is usually known in some way (e.g., the minimum is that we assume they exist, that we know that *we do not to know*) but we still persist to assume (common in statistical learning) that it is *unknown* (particularly when forming hypotheses or building unsupervised methods).

Ignorance is also a kind of inferred knowledge. Since inference is a process, it can be associated with common properties of processes, it can be intentional/unintentional etc. (like what we had for semantic roles)

Classification: Introduction: Inference (contd.)

and no mistake, Inference can be:

* rational or irrational;

* its result can be appended to our belief system or discarded;

* it can be accomplished using logic or other frameworks

* they can be modified or altered (e.g., look at the inference systems in Physics).

* truthfulness of its result can be graded and fuzzy or Boolean and decisive;

* inference mechanisms can be plugged into each other;

and so on.

* Inference can be <u>classified</u>, e.g., as Induction, Deduction, Abduction etc.

Classification: Introduction: Inference (contd.)

For our purpose, an inference mechanism is an algorithmic (or at least somehow reproducible) use of a knowledge base (and perhaps users' feedback) to do inference.

We often focus on inference that results in an outcome which is useful for us.

- E.g., traders make money using program-trading, a feature-based automatic inference system for trading stocks.
- Traders use program-trading only if it helps them to achieve their goals (which is often a lot of money in a short amount of time).
- In other words, money and time are the two decisive factor for most program-trading algorithms.

Classification: Introduction: Inference (contd.)

More precisely, an inference process that can be stated formally (using a meta-language), and its results can be used to serve a purpose and its usefulness can be measured in one way or another.

Simply put, at the edge of machines we need mechanized inference that can be evaluated (evaluation is important to be able to assess their usefulness – sorry if I restate an obvious matter).

Classification methods implement inference mechanisms, which are utterly simple to grasp and use, and yet very effective for solving complex problems.

Today we learn how to start making a classification system using pen and paper, from an intuitive perspective and without using any sophisticated mathematical formula.

Classification is not only a computer thing!

Classification is, just, natural and known to our brains (even in the distributional/statistical way that we will use it later).

The principle is simple: We associate items of thought (units of meanings) to each other.

* The item, which is described by other items is called a record (in our slides), but aka an observation, a random variable, vector, tensor, item, etc..

* The item that is used to describe the *record* is called a feature (in our slides) but also (else where) attributes, properties, etc..

Classification is not only a computer thing! (contd.)

NOTE: Records and features can refer to the same or distinct collection(s) of items.

For example: You can describe people by their associations to other people (record and feature both coming from the same collection of people). Or, you can describe people by their facial features, height, weight, race, etc. (record and feature are coming from two distinct sets) – this will be more clear later, and after all, it may not matter much.

Disregarding of the name we associate to a record, a record is always an structured representation of items/things/unit-of-meanings/thought, and it contains certain amount of information about them (encoded through feature associations).

Classification is not only a computer thing! (contd.)

Remember: RECORD is a structured REPRESENTATION.

Explanation:

NB: records are structured in the sense that they are quantified based on the features (this does not mean organized or directly measurable) used to describe them.

For instance, a brainwave signal (a record of unstructured data) is quantified based on the output of say 128-channel EEG caps (so something seemingly unstructured has a structured presentation).

A record is one sign (signature) of an item, an item can have an infinite number of signatures at any moment.

A Little Game!
Classification is not only a computer thing!

We can check this by a simple game (not so much related to us but I think it is helpful):

On a piece of paper write *five* words or simple phrases (for the sake of time) that comes to your mind when you hear these three concepts (let's tame our imagination and say that these concepts are people and we want to describe them):

A Little Game!
Classification is not only a computer thing! (contd.)

* TEA1: Your first teacher ever,
* TEA2: The teacher the course before this course, and,
* TEA3: The teacher of this course.

If you like, you can keep an electronic format of your words, in a format like:

nick-name TEA1—TEA2—TE3 word1 word2 word3 word4 word5

A Little Game!
Classification is not only a computer thing! (contd.)

Now this time think of the *Best Teacher* you ever had, and write 5 words that come to your mind.

(the experience of recalling all the TEAs and the best teacher may involve images and mute sounds, not always words, and not always easy to write them down).

A Little Game!
Classification is not only a computer thing! (contd.)

First, use your intuition and compare your *best teacher* and the TEAs: TEA1, TEA2, TEA3 (please write the result, also some information on how you did it).

Now, compare the *best teacher*-RECORD (i.e., the one you built using word-FEATURES) with the RECORDS of other TEAs. (please write the result, and some description on how you did your comparison).

A Little Game!
Classification is not only a computer thing! (contd.)

Checking:

Do you agree that our brains build some associations between these meanings (TEAs) and some word/images/brain-stimulation/etc. which we call them features.

A Little Game!
Classification is not only a computer thing! (contd.)

My **intended conclusion** is that things can be described by other things.

When put together, these Thing-by-Thing representations (*record-by-feature*) form a feature-based representation system that can be used for inference and reasoning.

Note that record-by-features must be built based on the same rationale/principle, this rationale maybe enforced by the representation system itself (e.g., constraining features by introducing a meta-vocabulary, etc.).

A Little Game!
Classification is not only a computer thing! (contd.)

Next step: Can you categorize, or label your features in the TEAs-by-words experiment? (i.e., to assign each word a label).

for instance, I would turn a record such as

bq TEA1 fun kid game dictation classroom

to

bq TEA1 feels:fun things:kid activity-l:game activity-h:dictation place:classroom

A Little Game! Classification is not only a computer thing! (contd.)

Now, throw out the old words and compare the TEAs using the labels you assigned to words.

E.g. instead of

bq TEA1 fun kid game dictation classroom

to

We would choose to use

bq TEA1 feels things activity-l activity-h place

A Little Game!
Classification is not only a computer thing! (contd.)

This time, use the new feature representation to compare the TEAs (i.e., the labels of words instead of the words).

Is the result different than when you are using words? How did you compare the records? Is your comparison strategy the same as the first experiments?

NB: SAME INFORMATION (words came to mind) BUT DIFFERENT FEATURES!

A Little Game! Classification is not only a computer thing! (contd.)

Could you categorize your TEAs to fav and unfav TEAs, and place these labels in front of your records?

e.g.,

fav: feeling-p things activity-l activity-h place

unfav: feeling-n age topic feeling-n place

unfave: things feeling duty feeling duty

fav: activity-h feeling-p feeling-p activity-l feeling-n

A Little Game!
Classification is not only a computer thing! (contd.)

You still have four records, each labeled by a class label fav and unfav.

Let's say you want to compare fav and unfav teachers? How would you do that using your records?

NB: NOTICE how we changed TEA records to fav/unfav records (class labels instead of TEA1, TEA2, . . .).

A Little Game!
Classification is not only a computer thing! (contd.)

Now, ask the person next to you to kindly provide you with their records for 3 random TEAs with the class label fav/unfav attached to them.

Ask for the remaining fourth TEA with **no** class label.

Can you predict the class label of the last TEA? (example is in the next slide)

A Little Game!
Classification is not only a computer thing! (contd.)

That is, you will be given the training data (TEA records of known class label):

unfav: feeling-n age topic feeling-n place

unfave: things feeling duty feeling duty

fav: activity-h feeling-p feeling-p activity-l feeling-n

and, you are required to predict **?!?!** in the test data:

?!?!: feeling-p things activity-l activity-h place

A Little Game!
Classification is not only a computer thing! (contd.)

So, what is **?!?!**, fav or unfav? Can you guess that? If yes, can you explain why you predicted **?!?!** as **fav** or **?!?!** (the so-called class labels)?

A Little Game! Classification is not only a computer thing! (contd.)

This is the last part of our game: Could you 'transpose/transform' your record-features to feature-records. E.g., this comes up for me (based on a number of assumptions):

feeling-n unfav unfav fav
feeling-p fav fav fav

* How many records do you have now?
* Do you assume each feature label is an item? Or, do you assume that a record per each occurrence of your feature?

What do you see there?! E.g., any correlation between the "transposed vectors".

A Little Game!
Classification is not only a computer thing! (contd.)

Did we do classification? Certainly yes, in a number of ways. Could you identify the common patterns in what we did when building record-by-feature representations?

I hope this little game gives a simple tangible idea of what we call classification.

Once you get this general idea, we are ready to formalize what we did intuitively and informally using a mathematical framework.

Classification: Introduction: Inference

We still have not touched the subject of classification in machine learning but hopefully we all have a good idea about classification process. Let's iterate once more what we know so far:

We can devise a record-by-feature representation system, which may be used for some type of decision making (reasoning/inference/etc.).

Our record-by-feature representations can be altered dramatically based on what:

* Records that we choose to describe for a purpose, i.e., various abstractions of TEAs in our game.

Classification: Introduction: Inference (contd.)

* And, based on the features (things) that we choose to describe records:

* We constantly changed our TEA-by-word representation to TEA-by-gword, FAV/UnFAV-by-gword, gword-by-FAV/UnFav, gword-by-TEA and so on....

* Of course, this can be altered to, e.g., a phrase-by-SyntaxSemanticFeat representation which can be processed and used for semantic role labelling.

Classification: Introduction: Inference (contd.)

As a side note, remember that records in our feature-based representation system can be used with different inference methods and strategies:

We can do induction, deduction, abduction, Bayesian inference, discriminative classification, hierarchical Bayesian inference, example-based (case-based reasoning), deep neural network CNN RNN GRU, whatever!!!

Classification: Introduction: Inference (contd.)

But I guess you all agree that doing our experiment on papers was not a) that easy (paper, pen, etc.), and b) it gets really messy soon.

We have, in a way, a structured representation but still it lacks some formal/mechanical representation.

Most people find tabular presentation suitable for the sort of task we did.

For instance, the rows will records and the columns will be features, and each cell will be filled by some information table. Agree?!

The form of the table will be discussed in length next week: in particular what goes in the table cells (e.g., a cell can even contain another table, right?).

Classification: Introduction: Inference (contd.)

And after all, how to store, use, and convert this records in computer programs for machine learning algorithms?

Simply put, the conversion of raw record-by-feature representations to representations suitable for a learning algorithm.

What would you do in terms of data-structure, mathematical model, and so on . . . .

What is next?

Next two sessions, we formalize the whole thing in a few simple steps:

* Hypothesis formation;
* Collecting record by features;
* Formal Representation (presumably a vector space, or contingency table);
* Possibly feature learning, weighting, and dimension reduction;
* Mathematical inference (perhaps involving training a learning model);
* Evaluation (hypothesis testing).

in a way accessible to people with humanities background.

Syntax and Parsing: Example of Dependency Annotations

In the following slides, I show you a dependency parse of the sentence in a CoNLL-U format.

Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

Could you please pay attention to the syntactic relationships between `Pierre Vinken` and the verb `join`.

Similarly, could you look at relationships between `board` and `join`.

CoNLL-like Syntactic Representation

1	Pierre	Pierre	PROPN	NNP	_	2	compound	_	_
2	Vinken	Vinken	PROPN	NNP	_	9	nsubj	_	_
3	,	,	PUNCT	,	_	2	punct	_	_
4	61	61	NUM	CD	_	5	nummod	_	_
5	years	year	NOUN	NNS	_	6	nmod:npmod	_	_
6	old	old	ADJ	JJ	_	2	amod	_	_
7	,	,	PUNCT	,	_	2	punct	_	_
8	will	will	AUX	MD	_	9	aux	_	_
9	join	join	VERB	VB	_	0	root	_	_
10	the	the	DET	DT	_	11	det	_	_
11	board	board	NOUN	NN	_	9	dobj	_	_
12	as	as	ADP	IN	_	15	case	_	_
13	a	a	DET	DT	_	15	det	_	_
14	nonexecutive	nonexecutive	ADJ	JJ	_	15	amod	_	_
15	director	director	NOUN	NN	_	9	nmod	_	_
16	Nov.	Nov.	PROPN	NNP	_	9	nmod:tmod	_	_
17	29	29	NUM	CD	_	16	nummod	_	_
18	.	.	PUNCT	.	_	9	punct	_	_

Syntax and Parsing: Example of Dependency Annotations

This time, we look at the SDP (semantic dependency parsing) records of the same sentence, in which semantic roles are listed in a number of columns.

For the ease of locating the entries, the Actor-argument, and the Patient-Arguemnt roles are highlited:

Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.

Semantic Dependencies Parsing Dataset

1	Pierre	Pierre	NNP	-	-	_	NE	_	_	_	_	_
2	Vinken	vinken	NNP	-	+	_	_	_	_	ACT-arg	_	_
3	,	,	,	-	-	_	_	_	_	_	_	_
4	61	61	CD	-	-	_	_	RSTR	_	_	_	_
5	years	year	NNS	-	+	_	_	_	EXT	_	_	_
6	old	old	JJ	-	+	_	DESCR	_	_	_	_	_
7	,	,	,	-	-	_	_	_	_	_	_	_
8	will	will	MD	-	-	_	_	_	_	_	_	_
9	join	join	VB	+	+	ev-w1777f1	_	_	_	_	_	_
10	the	the	DT	-	-	_	_	_	_	_	_	_
11	board	board	NN	-	-	_	_	_	_	PAT-arg	_	_
12	as	as	IN	-	-	_	_	_	_	_	_	_
13	a	a	DT	-	-	_	_	_	_	_	_	_
14	nonexecutive	nonexecutive	JJ	-	-	_	_	_	_	_	RSTR	_
15	director	director	NN	-	+	_	_	_	_	COMPL	_	_
16	Nov.	nov.	NNP	-	+	_	_	_	_	TWHEN	_	_
17	29	29	CD	-	-	_	_	_	_	_	_	RSTR
18	.	.	.	-	-	_	_	_	_	_	_	_

Can you devise a record for the semantic roles?

To go forward with our semantic role labeling task, we must build some feature vectors, or record-by-feature, representations.

Let's say we want to represent the semantic roles in the previous slides. Could you use syntactic parses to come up with some features for the agent and patient?

What are them?

A question to answer next week: To identity Actors and Patients, for which words do we need to build feature representation? E.g., those only with a direct syntactic relation? Do we need to consider irrelevant words, too? And why?

Bibliography

Palmer, M., Gildea, D., and Xue, N. (2010). *Semantic Role Labeling*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

We can

We can collect our TEAs data in a shared spreadsheet (using pseudonyms) and see what comes out next week.

We can also look at a few basic resources and material, if you want to do excesses:

* CoNLL-U data format
`http://universaldependencies.org/format.html`

* Also, the data format presented at (you do not need the license, just download the public toy data)
`https://competitions.codalab.org/competitions/19159#learn_the_details-datasets`

* Alternatively, familiarize yourself with constituent parse trees which are used in Chapter 3 of Palmer et al. (2010) (see Prof. Kallmeyer's parsing course).