On Chomsky and the Two Cultures of Statistical Learning

This paper discusses probabilistic and statistical language models in a favourable manner, as a direct retort to Professor Noam Chomsky's statements at MIT's *Brains, Minds, and Machines* symposium, which ridiculed researchers using purely statistical methods to solve various linguistic problems. Chomsky acknowledges the engineering successes that statistical methods have had, but firmly believes that these methods are "irrelevant to science" and "provide little insight" to humans actually studying these linguistic problems. He claims that language generation could not possibly be done accurately by statistical methods - as people don't decide what the third word of a sentence will be based on a probability keyed from the first two. However, he fails to see the great strides that statistical models have made in the area of language interpretation, which can be seen as an inherently probabilistic issue - in terms of speech recognition, what is the most likely meaning of some noisy input? Linguistic tradition, which Chomsky has been a big part of creating, has favoured models that are discrete, categorical, and qualitative, which new statistical models tend not to be.

Following this discussion of Chomsky, Norvig introduces the idea of the *two cultures* of statistical learning, which are claimed to be the *data modelling culture* and the *algorithmic modelling* culture. The first of these ideas says that nature can be described as a black box, mapping input to output with a relatively simple unknown model, while the second holds the idea that nature cannot possibly be described by a simple, unseen model. Chomsky seems to hold issue mostly with this second category, as these sorts of models tend to produce output that is not easily interpretable, and doesn't make any attempt to explain *why* something happens, just that it *does* happen. These statistical models tend to make linguistics into an empirical science rather than a mathematical one, which counteracts Chomsky's philosophy that people should focus on *why* things are the way they are, and that mere explanations of reality aren't important.

Languages are complex - often seeming random and ever-changing - and Chomsky believes they could never be accurately described by an entirely statistical model. Yet probabilistic reasoning has been continuously useful in modelling how we use language. Chomsky's idealistic, mathematical idea of language is also a system that would be hard for humans to interpret and it also misses the core point of how language works.

Significance

This text provides a great introduction to the many schools of thought about language modelling, as well as a well-argued critical approach to the work of one of the world's most influential linguists.

Limitations

This text discusses the inherent limitations of both purely statistical and purely mathematical linguistic models, neither of which can be used in isolation to fully describe the complexity of language, and is limited itself by not providing a suitable alternative.

Contextual Word Representations: A Contextual Introduction

This text provides a broad introduction to natural language processing (NLP), an ever-growing branch of artificial intelligence and computational linguistics, that can be applied in a range of instances, such as text interpretation and translation. Firstly, a few definitions:

Word token:

- Individual words observed in a piece of text.
- Easy to determine for analytical languages such as English, but more difficult with agglutinative languages that join words together.

Word type:

- Distinct word token
- Two instances of the same word token share a word type

Word types can be represented as a string, but comparing strings is costly in terms of time, so strings could be *integerized*, meaning each word type could be assigned to an arbitrary integer. As these types are stored arbitrarily, they are entirely **discrete**, and cannot be compared for anything other than direct identicality. However, words in a language can clearly be categorised in a number of ways, broadly by classifying *parts of speech*, or more narrowly by identifying individual categories within these groups. This idea that two non-identical words may be more or less similar is integral to NLP, and is infinitely useful when applying machine learning methods, in order to exploit similarities between words so the algorithm can generalise. This new idea of word types could be defined as a **vector**, giving us the opportunity to assign dimensionality in order to form connections between words. These connections (features) can be assigned by algorithm or by hand, and can vary widely in scope.

Examples of the features that word vectors can contain:

- Assignment of a binary value 1 to collections of word types that belong to a given class and 0 to those that don't belong to that class.
- Word types that are variants of the same underlying root can be marked in a similar way (words like *did*, *done*, *doing*, *and do* can all be classified together)
- Word types that could be mapped to magnitudes or directions could be classified together (associating weight or size to particular nouns)

Words and expressions that can be applied in similar situations are likely to have related meanings. **Clustering** is a technique that can be used to automatically derive features based on this idea, by taking a distributional view of the many contexts a word appears in. Words that appear in similar contexts can be clustered together and added into a hierarchical cluster tree which can be applied in a range of functions. One issue with these word vectors is that they can contain a lot of information, much of which may be useless except in very specific circumstances. Simpler vectors can be created through dimensionality reduction, and can actually help to reduce corpus-specific noise as well as being more efficient to use.

Methods for obtaining distributional word vectors:

- Large neural networks can be fed pretrained or predetermined word vectors which can be treated like parameters and adapted to vectorise new corpora.
- Bilingual corpora can be used to align word vectors in two languages into a single vector space, in order to get a more complete picture of potential word contexts.
- Word vectors can be calculated based on character similarity, meaning words from languages with highly–inflectional morphology can be categorised together. This can also be useful for categorising alternate variations of English words (e.g. simplifications and abbreviations used in text messages).

Since many words can have unique meanings in separate contexts, these word vectors must often be contextual, containing different senses for many words. Word token vectors can be used over word type vectors for this, capturing only what a word means in a specific context. *Embeddings from language models* (ELMo), has made advances in the field of word token vectors by creating arbitrarily long 'context vectors' of words which appear on either side of a given word token. ELMo has been effectively used to answer questions on content when given a paragraph, as well as being very good at labelling both semantic arguments of verbs and expressions of proper nouns.

Word vectors unfortunately suffer from the same inherent biases that have plagued machine learning algorithms for years, as they can only be as impartial as the data they are trained on, meaning these corpus-derived word vectors can tend to associate certains job titles with certain pronouns or use to unwittingly employ derogatory terms. Contextual word vectors do offer some new possibilities to avoid overgeneralisation of distributional patterns but they are still far from perfect.

Significance

This paper supplies a thorough introduction of how we handle words and context in NLP, and how new word vector methods can be used to improve language modelling systems.

Limitations

Language, and therefore NLP, deals with much more than just words, and even modern contextual word vectorisation cannot fully capture how language is actually used and how it can rapidly evolve.

A Question

Is it really possible that we'll ever make a language modelling system that could be better at using language than a human?