Deep learning of text understanding: Google's BERT

Professor Eric Atwell: Hello, In this lecture, we're going to be looking at BERT. BERT is a Bidirectional Encoder Representations from Transformers. A method and tool kit and dataset and resource from Google Research Labs for understanding meaning relationships between sentences or rather understanding measuring the similarity in meaning between two sentences. It can be used for many tasks, which involve pairs of sentences which are related in some way. And the reference for the paper by Jacob Devlin and others. BERT, pre-training of Deep Bidirectional Transformers for Language Understanding. Notice, it got the best paper award at the 2019 conference NAACL. That's the North American Chapter of the Association for Computational Linguistics, the main professional body for computational linguistics researchers. And notice also, this is not just academics in universities but also in research labs. So Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova are all from Google's AI Language Research group. And it makes you think, perhaps, university research teams can't really compete nowadays in AI research, and in particular, in computational linguistics research, with the likes of very large and well-funded research labs at Google, Microsoft, Facebook, Amazon, and so on.

However, because Google has this aim to make things useful to everybody, we have a benefit that they tend to make these computational linguistics resources freely available for other people to use. OK, so here's the overview of what's coming up. First of all, the general idea of BERT is that you have some pre-trained neural network representations learned from unlabelled text, in principle, unlabelled text. And then once you've got this language model, general language model of English, then you can fine-tune it for specific tasks like answering questions or translating sentences into another language. And the clever thing, one of the clever things about BERT is it does a lot of unsupervised learning of morphology, that's, word structure within words, syntax or grammar, and semantics or meaning. Unsupervised in the sense that there isn't a lot of information in terms of data, which is already labelled with the correct answers. But we'll see that they've cheated in a way.

In a minute, we'll see that it's basically unsupervised learning. And at the heart of this, in terms of learning word segmentation, they use a method called WordPiece, which is unsupervised machine learning of text segmentation into pieces of word. Words and pieces of word, I should say. And then they predict sentence meaning from WordPiece meanings by joining together the WordPiece meanings, and they assume that in running text, two sentences next to each other should be related in meaning and that basic assumption allows them to take, pluck, pairs of sentences which they have related meaning, and then take one sentence and another sentence at random from the corpus, and assume it's not related meaning. And that gives them labelling, so the results are the general model for measuring the distance between two sentences. And then, this could be fine-tuned on specific human-labelled natural language processing datasets to learn the specific task of that dataset. And then they made a lot of experiments and lots of tables of results, which basically show that both BERT Base, a little version of BERT, and BERT Large, the much larger neural network, outperform all the systems on all tasks by a substantial margin. And for the same pre-trained model can successfully tackle a broad set of natural language processing tasks. And we'll see some of this, and the paper also, and it's got a lot packed into it.

It's also got ablation studies. What happens if you deliberately break bits of BERT and see how well it works in disabled mode? It's also got a huge set of references, some very detailed appendices of lots more information. And then finally, I'm going to finish with some questions about what isn't in the paper, which we might want to look at. OK, so there's a lot coming up in a nine-page paper plus some appendices and references. OK, so the first idea overall is that BERT is about learning pre-trained representations from large amounts of unlabelled text, and then this representation is a representation of whether or not two sentences are related in meaning, and then you can fine-tune it for very specific tasks.

So BERT stands for Bidirectional Encoder Representations from Transformers, so the transformers are the neural networks, and the representations, the things you learn, so they are essentially data and not software. So don't think you can download the BERT.pui program and then run it. Although actually, there is a BERT program as well.

So if you go to the website, they do have software as well, which uses the BERT models or trains the BERT models. The next idea is of pre-trained representations from unlabelled text. Now the idea is that it's a general model of English word structure and grammar. And for pre-training, they use a huge corpus, 800 million words from the Google Books corpus. If you go to books.google.com, this is a very large collection of books scanned in from Oxford library, Oxford University library and many other University libraries. So there's a very large collection, mainly learned texts, but actually anything that's in university libraries. And also, English Wikipedia, which is an even larger and growing collection of English language text. So it's English.

It can be fine-tuned to create models for a wide range of tasks. And tweak for specific language tasks which involve pairs of sentences in some way. And for state-of-the-art results on 11 natural language processing tasks. They give quite a few examples of specific tasks, and each of them they do very well. Now, the code and the pre-trained models are all available from the Google Research BERT GitHub website, so you can have a look yourself. Beware that the method for learning a model takes very large amounts of processor and memory. But once the pre-trained model, that they once have learned it, that can be plugged into simpler tasks, and you can run it on a laptop, if you don't mind waiting a few hours, or some time, let's say.

OK, so the core linguistically about this is unsupervised machine learning. Unsupervised machine learning of morphology, that's word structure, syntax and grammar, or semantics as well. So it's a big neural network, a transformer neural network. So if you do a deep learning course, you may learn something about transformers as a particular structure for neural networks. I'm not going to go into details here. There is some more detail in the Jurafsky and Martin textbook. But essentially, it's a way of having inputs and outputs, and mapping not just a sentence into a class, as a classifier might do, but a pair of sentences and a class.

So in training, you show BERT many examples of sentence and sentence, and the result is yes or no. And, yes, are they related in some way? And test is also given a new sentence plus another sentence. BERT will predict yes or no. So BERT is learning sentence plus sentence plus class, the yes or no, whether or not they are related. Without any linguistic theory in the model, there's no explicit rules about word structural morphology, grammar or syntax, or even meanings. Instead, you use a range of unsupervised or semi-supervised machine learning models for morphology, syntax, and semantics from a large text corpus. This should remind you of the Morpho Challenge you've heard of before, which was unsupervised learning of morphological analysis, segmenting words into morphemes. Originally it was developed for highly complex languages like Finnish or Turkish, where words could be quite long, consisting of several morphemes. But this also works for English.

OK, and you also should remember word2vec. This is an unsupervised measure of learning semantic vectors for meanings of words or morphemes. The idea being that the meaning of a word if you have a reasonably common word, like cat, you look up thousands of examples of cat in a corpus, and take the words on either side. The context of the word cat, and those contexts are words which appear and essentially those words give you the meaning of cat. So the meaning of cat is similar to the meaning of dog if the context vectors of cat are similar to the context vectors of dog. And you also, on top of that, you have this unsupervised learning of sentence pairs being similar meanings, by giving lots of pairs of sentences which are similar, and then taking random pairs of sentences which are not similar, and then it learns to distinguish between similar or not similar.

OK, so the first stage in all this is WordPiece, unsupervised machine learning of text segmentation into words and morphemes. First of all, the first stages to segment each of the sentences into bits, but not words necessarily. So linguistic theory tells us that there are words and morphemes, that words is made up of word plus plural s. That is a root and an affix. But that's just the theory, that only works if you have a dictionary, and that works for English, we have lots of dictionaries for English. But Google is very keen on methods which work on any language, even if there aren't lots of resources for it already. And that's why the Morpho Challenge was aiming to do Finnish and Turkish and other such languages because there aren't that many formal language learning linguistic resources for all these other languages. So that's one issue.

Another issue is that in any corpus, there's always going to be more and more new words you haven't seen before, so how do you deal with those? There's rare words which occur very occasionally, and there's lots of out of vocabulary words, words you haven't seen before. Well, what WordPiece does, this is inspired by some other Google researchers working on machine translation, and in machine translation they did have to handle lots of languages where they didn't have as many resources, as for English. So WordPiece works pretty much for any language which is written in some character-based method, and you set in advance how many WordPieces you want. What is the size of the vocabulary? And 30,000 seems to be a reasonable number. It turns out that a good English dictionary may have 30 to 50,000 entries in it. Base words or roots in it. So that's a reasonable number.

So WordPiece is an unsupervised machine learning morphological analyser. What it does is it takes the entire corpus, takes out all of the words, and it tries to compute frequencies of words and pieces, such as, the common words are kept as pieces because they're quite frequent. For all the less common words, you try to chop it up into pieces such that the pieces are also common. So, for example, unsupervised may be quite rare, but if you chop it up into un, supervise, and duh, then supervise is quite common, and un is definitely common, and duh is also very common. So all the rare words are chopped into smaller parts, and then it tries to work out-- it tries essentially various possible permutations and combinations until it finds the best entropy combination, the one which has most reasonably low-frequency pieces.

So this is reminiscent of Morpho Challenge, it wasn't part of the Morpho Challenge, but they did it separately at Google research themselves. So there's no more rare words or out of vocabulary words. There's only common word pieces. And for each WordPiece then just as in the style of word2vec, you input the text, only this time instead of dividing up into words, you divide up into WordPieces. And then, you build a vector representing the concordance context for each WordPiece. What are the words or WordPieces which appear before and after it? And that gives you a vector of frequently occurring concordances or contexts. OK, so now we have a way of representing all the words, even the rare words because they're made up of WordPieces, and then we have to predict sentence meanings. The meanings of the whole sentence. Not just individual words. Well, it's not explicitly stated in the paper, but if you look into the code, you should find that the default assumes it's about 25 WordPieces per sentence. And if there's less than 25, well, sorry, if there's more than 25 words, then the default is just to ignore the rest of the sentence. Or if there's less than 25, so you fill it in with blanks. But that's surprisingly, surprisingly to me as a linguist, that seems to work OK. Anyway, then, so the semantics of a sentence is essentially joining the word meanings of each of the WordPieces, the vectors of each of the WordPieces, but joining in a rather sophisticated way. You don't just simply add them together or concatenate them.

We do not use traditional left-to-right or right-to-left language models to pre-train BERT. We simply mask some percentage of the input tokens at random and then predict those masked tokens. This is a Masked Language Model, although it is often referred to as a cloze task So if any of you have learned English as a second language, you might have come across cloze tasks. This is a test of your English language ability. You're given a sentence, and one of the words is blanked out, and you have to predict what the word is. That's essentially what BERT tries to do. And therefore, it optimises the vector representing the meaning in terms of how good the best vector is predicting the missing word. So that you have a vector for all of the sentence, and then you take out one of the words or some of the words, and you see how you can optimise the vector to predict those taken out words. And this has to do with the details of a neural network, if you really want to find out more, there's a better explanation in the Jurafsky and Martin textbook, and you could try this with the code from the web page.

OK, so then they've got a method then for representing meanings of-- well, first of all, for the chopping up the sentence, the text into WordPieces, and then representing the meaning of each WordPiece, and representing the meaning of a sentence as a combination of the WordPieces. Now, the next task is how do you represent relationship between two sentences? Well, they came up with a really neat idea that you can assume that in any coherent text, each sentence follows on from the previous sentence.

If I'm writing a sentence, or writing some PowerPoint slides, or in Wikipedia, or in Books given that their training set was Wikipedia and Books, each of these documents, each Wikipedia entry is actually not just a collection of sentences at random, but it is first sentence followed by second sentence followed by third sentence followed by fourth sentence. And there is a meaning relationship. What the meaning relationship is a bit vague, but they are related in meaning. Otherwise, the whole text wouldn't make sense. So they came up with this idea for each sentence is related meaning to the next sentence. It's a bigram model for sentence meaning. So this is a way of using the text without actually explicitly labelling it. It is sort of human-labelled. It does make the assumption that the author labelled each sentence, each sentence as being related to the previous sentence. And then, if you take a sentence and another sentence at random from somewhere else in the corpus, then it's not related because it hasn't got the human label if you like. There's an implicit label between sentence one and sentence two saying they are related. An implicit label between sentence two and sentence three saying they are related in some way. So they're able to pre-train a Next Sentence Prediction task. In training sentences, A and B, 50% of the time, B is the actual next sentence that follows from A, in the corpus, labelled as, IsNext, in the Book, or in the Wikipedia entry. And 50% of the time, it is a random sentence from anywhere else in the corpus labelled as NotNext. So the input sentence A and sentence B, from the pre-training, are related by IsNext. But this is very similar to, analogous to, sentence pairs in paraphrasing. So when you have sentence B is a paraphrase of sentence A, means the same thing as sentence A. Or also, in hypothesis-premise pairs in entitlement. Sentence A entails sentence B. That's another relationship. Or in question-passages pairing, questions and answers, the question and the answer are related in a similar way as to, sentence one is related to sentence two. And even in text to zero pairs, in text classification.

If you want to have a text sentence, which is classified as A or B, but it's a bit like saying you've got a pair of sentences, but the second pair-- second in the pair is nothing. So all of these specific tasks are much like the general Next Sentence Prediction task. So you could take the general model and then fine-tune it by giving it an extra training set for this particular task. So it's all fed into an output layer for classification. So then, they went on to try this out on a number of hand-labelled natural language processing datasets and tasks collected by various computational linguistics researchers.

As I've said before, a lot of computational linguistics research is done in conferences. You typically have people presenting their own papers, but they also have these shared tasks, where the organisers have got together a standard hand-labelled classification training set and test set. And anybody can enter. And you're given the training set, and the task is, can you come up with a classifier which predicts correctly? The test set. And you're not actually given the answer to the test set, but you're evaluated on the test set. And GLUE is like a super competition. So the GLUE researchers got together a number of existing datasets and put them together into a general language understanding evaluation set of tasks. So they didn't actually have to do any labelling themselves they got together several different ones, so that anybody can evaluate on all of these things.

So SST-2 is a movie review. Given a movie review, can you predict is it positive or negative? Do they like the review or don't like the review? Well, CoLA is given a sentence. Is it linguistically acceptable? Yes or no? So both of these tasks have just got one sentence. So in BERT terms, you give a sentence and a null or blank for the second sentence, and the yes or no in the first sentence answers yes or no, is it positive? And the second one, yes or no, is it linguistically acceptable? So they're the same sort of tasks but slightly tuned from the original pair task. STS-B and MRPC are two other datasets. In this case, there are pairs of sentences, and the task is, are they similar in meaning? So this is much more like the original BERT training task.

There's also the QQP, which is a pair of questions on Quora. Are the two questions semantically equivalent? Now, Quora is a website where anybody can post a question, but before you post a question, you should see has anyone else already posted a similar question. So this would be a useful classifier to do that. This is actually very similar to STS and MRPC, except the two sentences are both questions.

Then we have QNLI, which is given a question and a sentence. Does the sentence contain the correct answer? So it's not really a question and answer, but rather is the answer within the sentence? Because sometimes you can have more than just the straightforward answer. But it is essentially question and answer in it's natural language inference. From the question, can you infer the answer?

Then we have two more. MNLI and RTE. Given the first sentence and a second sentence, is there an entailment or contradiction? So does the first sentence entail the second sentence, or does the first sentence contradict the second sentence? Or, as a third option, are they neutral? So each of these are, basically, is there a meaning relationship between a pair of sentences, but some variation on that? So those are all standard test sets within GLUE. And the Google researchers also tried to other quite large datasets.

There's the SQuAD dataset, Stanford Question Answering Dataset, as pairs of questions and answers. But in this case, it's a bit more complicated, given a question and the other text is a text span in a Wikipedia passage. Is the answer in the span of Wikipedia passage? So is this the Wikipedia text contain the answer, and where is it? And there's also SWAG, which is Situations With Adversarial Generations. Given a sentence, choose how it finishes from four choices. So both of these are a bit more complicated, there's a sentence, and then the second piece of text isn't just a sentence. In SQuAD, it's a larger piece of text, and you have to try to predict where in there is the answer.

And then SWAG, the second bit is four possible continuations and you've got to choose one of them. So it's a bit more complicated. But you can see that BERT, the underlying model, can be modified a bit or fine-tuned, as they put it, to fit any of these tasks. And they did try this, and they found that both BERT Base, the simple language model, and BERT Large, the larger language model, outperformed all of the existing systems on all of the tasks. They tried. So various other existing systems and BERT won by a substantial margin. And you can see the paper for what that margin actually is. And furthermore, BERT Large significantly outperforms BERT Base across all the tasks. It gets a better score. But it takes a lot longer.

In terms of neural networks, the Base has got 12 layers, and the Large has got 24 layers, so it takes a lot more processor, a lot more time, and don't try these experiments on your laptop unless you want to spend a long time waiting. That's the main thing. In addition, a paper is only nine pages long. So if you submit to a conference, like this, typically they invite short papers of four pages or long papers of eight pages. And if your paper is accepted, you're given an extra page to add in anything else that the reviewers asked you to add in, so nine pages is about maximum. You can also add in references not counted in the nine pages and maybe an appendix if you really need to. So this paper has got references down in the appendix. Incidentally, for coursework, do please stick to the page limits, so that doesn't apply, as is these extras don't apply to your coursework. Anyway, they have to have-- so in addition to showing the Bert model, and showing lots of results to say how good it is, they also tried some what they call ablation studies.

So this is a bit like in psychology, if you want to look at how the human brain works, then cut out bits of the brain and see what happens. You don't cut out bits of the brain but rather you study people who've had accidents, or for other reasons, parts of their brain doesn't work anymore and see how they behave and see what insights that gives you on normal brain behavior.

In the same way, in neural network studies, people like to turn off bits of a neural network and see how this affects performance. And there's more on this in the BERTology paper that you can read as well. OK, the conclusion is that deep bi-directional architectures, that's a neural network architecture, where you don't try to predict the next word or try to predict the previous word, but try to predict the whole sentence using this cloze test architecture. This allows the same pre-trained model to successfully tackle a broad set of NLP tasks. The BERT works for all sorts of tasks which involve pairs of sentences being semantic related in some way. They've also got quite an extensive set of references, so they build on lots of other research. For example, the WordPiece research wasn't part of this paper, but they were able to use that even though that of itself is quite a substantial novel piece of research.

So if you want to find out, or get some overview of the current state-of-the-art in natural language processing research, just read all the papers in the references. You'll find that's quite a lot there. We also have quite a lot of appendices with implementation details and also references to the website where you can download the software more details on the experiments and the ablation studies. OK, so they managed to pack a lot into nine pages, plus references, plus appendices. But even so, there are some issues which they haven't gone into some detail, so you might want to think about this. First of all, there's the implementation details.

If you actually want some code examples to try out yourself, then there is a URL, and you can go to the website and go to the Google archive there. They haven't really said something about-- this is a big issue in theoretical linguistics, is that long sentences don't mean the same as short sentences. I love you, obviously means something quite different from, I am very strong and bold, and I love you because you are tiny and unique. OK, but they haven't really discussed how they coped with different lengths. It simply says that they are input. A sentence is a combination of the word pieces. They also, they mention some of the computing resources required, but they haven't really said, and they did say, OK, to learn the underlying BERT model as particularly the Bert Large model requires huge computing resources. But once it's trained, then you can plug the pre-trained model into downstream tasks like question-answering, and this, to some extent, this sort of thing could be done on a laptop but you'll find they don't really talk about what bits of it work on smaller resources.

Another thing they don't really do is that they talk entirely about the overall accuracy of a language model. They don't give what linguists and corpus linguists like to see is specific examples of sentences or pairs of sentences which work and pairs of sentences which don't work. And explanations or some discussion or analysis of why it didn't work. So that's in the BERTology paper, there's more discussion about that. They haven't really talked about what tasks in NLP don't suit BERT. This isn't just their problem, but this is very common in research papers, where they present some results and then say our results are really good. And they could also be used for lots of other tasks. But they haven't said what tasks are not suitable for BERT. So BERT, inherently, is a language model based on sentences from Books and Wikipedia pages, and it assumes that one sentence is linked to the next sentence in some sort of meaning relationship, and it does work with other tasks, which are something like pairs of English sentences, which are readily related in meaning. But not all of natural language processing is about pairs of sentences which are related in the meaning.

It is also only about English, and they haven't even mentioned how this transfers to other languages. Having set this up and made the software available, other researchers have gone on to try it on other languages. So the example is Arab BERT, which is trained on a large Arabic text dataset and does the same thing with pairs of Arabic sentences. It assumes the WordPiece tokenisation, which is a clever way of capturing the meanings of common words, and then for rare and out of vocabulary words, they're broken up into pieces and, hopefully, the pieces are common. But it doesn't deal with multi-word expressions because the default assumption is that a piece is either a word or a bit of a word, but not a multi-word. And it doesn't. It's not clear how this works for languages other than English which are much richer morphology. Like Arabic, for example, the Arab BERT system also uses WordPiece, but maybe something other than WordPiece would be more appropriate for Arabic. OK, the final issue that doesn't really deal with much is how do we understand how BERT works?

For people like me who are trained in linguistics and computational corpus linguistics, at least, it seems counterintuitive that BERT can learn mapping between the sentence and sentence and class, yes or no, without any linguistic theory, any rules of morphology, any dictionary of words, and morphemes in English. Any grammar of English which says that words are nouns or verbs or adjectives or whatever, there's no semantic model which says that words can have more than one meaning, words have senses, or that two words can mean the same thing, there's nothing explicit about that. But there is in a dictionary. We just don't bother using a dictionary at all OK, so these are questions about BERT for you to think about, and if any of these questions grab your attention, then maybe you can use these as part of your research proposal. Or maybe for your project, later on, when you come to do a project, you could choose one of these questions as something you want to investigate further. This is well worth doing a longer project on.

OK, so in summary, we've looked at BERT as a model for pre-trained representations from unlabelled text. And unlabelled, at least, except for in the sense that the assumption is that there is an inherent, implicit label between each sentence, say that the sentence one is related to sentence two, and sentence two is related to a sentence three, and also that sentence one is not related to any other sentence at random plucked from the corpus. So those are implicit labels, at least so. And then the idea is that it's learned, it's trained to learn that the two sentences are related, and then it can be fine-tuned on specific relationships, like sentence one is the question and sentence to is the answer or sentence one is contradictory to sentence two.

OK, then we have, and it's fine-tuned by giving some training examples with that specific relationship, and then you test it on some test set with the same relationship. And it does unsupervised learning of morphology using WordPiece. Unsupervised learning of grammar and semantics, inherently, by this Word Piece, unsupervised learning of text segmentation. And then, it predicts sentence meaning by combining WordPiece meanings, but using this cloze test and model, and then it assumes adjacent sentences have related meanings, and then it can be fine-tuned on hand-labelled sentences, where the hand labelling says what is the relationship between the two sentences. The results are quite astounding, and this is why everyone is using BERT. That both BERT Base and BERT Large outperform all the other systems they tried on all of the tasks by a substantial margin. The same pre-trained model can successfully tackle a broad set of natural language processing tasks. And on top of all this, the paper also has some ablation studies looking at what bits of BERT are actually contributing what functionality? There's a large set of references giving you a broad reading for the current state-of-the-art research in natural language processing. There's some detailed appendices with explanations of the experiments in more detail. And finally, I presented some examiners some questions we're not so sure about BERT from the paper, at least, that you might want to follow up, in further research.

OK, I hope you've enjoyed this. Notice, in the summary of BERT paper, I've not talked much at all about the neural network architecture or the mathematical modeling. There's more of this in the Jurafsky and Martin textbook. But actually, probably if you use BERT, then you don't-- I mean, they built BERT as a black box. They made it available, at least, as a Black box that you can use on other tasks. And that the beauty is you don't necessarily have to understand the neural network model unless you want to become a machine learning researcher, developing variations of neural networks. So if you actually want to build new neural network architectures, then you have to do that. But for that, I would advise you to take on the deep learning course and get more on the deep learning architectures within that module.

OK, thank you for listening, and goodbye for now.

[END]