Word sense disambiguation

W4705: Natural Language Processing

Fei-Tzin Lee
(with slides from Kathy McKeown)

October 28, 2019

Announcements

- Kathy is away today, but will be back on Wednesday!
- Minor updates to HW3 - both assignment instructions and code

Today

- Word sense disambiguation, the classical way
- Word sense disambiguation... with BERT?

Outline

1. Word Sense Disambiguation
2. Contextual word embeddings and disambiguation

Word Sense Disambiguation

Announcements

- Today: Semantics
- Remainder of class: applications with MT first
- Two guest lecturers:
  - Kapil Thadani, Yahoo Research, Nov 6th
  - Or Biran, Elemental Cognition, Nov 20th

Word Sense Disambiguation (WSD

- Given
  - a word in context,
  - A fixed inventory of potential word senses
- decide which sense of the word this is.
  - English-to-Spanish MT
    - Inventory is set of Spanish translations
  - Speech Synthesis
    - Inventory is homographs with different pronunciations like *bass* and *bow*
  - WordNet senses

Two variants of WSD task

- Lexical Sample task
  - Small pre-selected set of target words
  - And inventory of senses for each word
- All-words task
  - Every word in an entire text
  - A lexicon with senses for each word
  - Sort of like part-of-speech tagging
    - Except each lemma has its own tagset

Supervised Machine Learning Approaches

- Supervised machine learning approach:
  - a training corpus of ?
  - used to train a classifier that can tag words in new text
  - Just as we saw for part-of-speech tagging, statistical MT.
- Summary of what we need:
  - the **tag set** (“sense inventory”)
  - the **training corpus**
  - A set of **features** extracted from the training corpus
  - A **classifier**

Supervised WSD 1: WSD Tags

- What's a tag?

WordNet

- http://www.cogsci.princeton.edu/cgi-bin/webwn

WordNet Bass

The noun ``bass'' has 8 senses in WordNet

1. bass - (the lowest part of the musical range)
2. bass, bass part - (the lowest part in polyphonic music)
3. bass, basso - (an adult male singer with the lowest voice)
4. sea bass, bass - (flesh of lean-fleshed saltwater fish of the family Serranidae)
5. freshwater bass, bass - (any of various North American lean-fleshed freshwater fishes especially of the genus Micropterus)
6. bass, bass voice, basso - (the lowest adult male singing voice)
7. bass - (the member with the lowest range of a family of musical instruments)
8. bass -(nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

Inventory of sense tags for *bass*

WordNet Sense	Spanish Translation	Roget Category	Target Word in Context
bass^4	lubina	FISH/INSECT	. . . fish as Pacific salmon and striped **bass** and. . .
bass^4	lubina	FISH/INSECT	. . . produce filets of smoked **bass** or sturgeon. . .
bass^7	bajo	MUSIC	. . . exciting jazz **bass** player since Ray Brown. . .
bass^7	bajo	MUSIC	. . . play **bass** because he doesn't have to solo. . .

Supervised WSD 2: Get a corpus

- Lexical sample task:
  - *Line-hard-serve* corpus - 4000 examples of each
  - *Interest* corpus - 2369 sense-tagged examples
- All words:
  - **Semantic concordance**: a corpus in which each open-class word is labeled with a sense from a specific dictionary/thesaurus.
    - SemCor: 234,000 words from Brown Corpus, manually tagged with WordNet senses
    - SENSEVAL-3 competition corpora - 2081 tagged word tokens

Supervised WSD 3: Extract feature vectors

- Weaver (1955)
- If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words. […] But if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then if N is large enough one can unambiguously decide the meaning of the central word. […] The practical question is : ``What minimum value of N will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?''

- dishes
- bass

- washing *dishes*.
- simple *dishes* including
- convenient *dishes* to
- of *dishes* and

- free *bass* with
- pound *bass* of
- and *bass* player
- his *bass* while

- “In our house, everybody has a career and none of them includes washing *dishes*,” he says.
- In her tiny kitchen at home, Ms. Chen works efficiently, stir-frying several simple *dishes,* including braised pig’s ears and chcken livers with green peppers.
- Post quick and convenient *dishes* to fix when your in a hurry.
- Japanese cuisine offers a great variety of *dishes* and regional specialties

- We need more good teachers – right now, there are only a half a dozen who can play the free *bass* with ease.
- Though still a far cry from the lake's record 52-pound *bass* of a decade ago, "you could fillet these fish again, and that made people very, very happy." Mr. Paulson says.
- An electric guitar and *bass* player stand off to one side, not really part of the scene, just as a sort of nod to gringo expectations again.
- Lowe caught his *bass* while fishing with pro Bill Lee of Killeen, Texas, who is currently in 144th place with two bass weighing 2-09.

Feature vectors

- A simple representation for each observation (each instance of a target word)
  - Vectors of sets of feature/value pairs
    - I.e. files of comma-separated values
  - These vectors should represent the window of words around the target

  How big should that window be?

Two kinds of features in the vectors

- **Collocational** features and **bag-of-words** features
  - **Collocational**
    - Features about words at **specific** positions near target word
      - Often limited to just word identity and POS
  - **Bag-of-words**
    - Features about words that occur anywhere in the window (regardless of position)
      - Typically limited to frequency counts

Examples

- Example text (WSJ)
  - An electric guitar and **bass** player stand off to one side not really part of the scene, just as a sort of nod to gringo expectations perhaps
  - Assume a window of +/- 2 from the target

Examples

- Example text
  - An electric guitar and **bass** player stand off to one side not really part of the scene, just as a sort of nod to gringo expectations perhaps
  - Assume a window of +/- 2 from the target

Collocational

- Position-specific information about the words in the window
- guitar and bass player stand
  - [guitar, NN, and, CC, player, NN, stand, VB]
  - $Word_{n-2,}$ $POS_{n-2,}$ $word_{n-1,}$ $POS_{n-1,}$ $Word_{n+1}$ POS_{n+1}…
  - In other words, a vector consisting of
  - [position n word, position n part-of-speech…]

Bag-of-words

- Information about the words that occur within the window.
- First derive a set of terms to place in the vector.
- Then note how often each of those terms occurs in a given window.

Co-Occurrence Example

- Assume we've settled on a possible vocabulary of 12 words that includes guitar and player but not and and stand

- guitar and bass player stand
  - [0,0,0,1,0,0,0,0,0,1,0,0]
  - Which are the counts of words predefined as e.g.,
  - [fish,fishing,viol, guitar, double,cello…

Classifiers

- Once we cast the WSD problem as a classification problem, then all sorts of techniques are possible
  - Naïve Bayes (the easiest thing to try first)
  - Decision lists
  - Decision trees
  - Neural nets
  - Support vector machines
  - Nearest neighbor methods…

Classifiers

- The choice of technique, in part, depends on the set of features that have been used
  - Some techniques work better/worse with features with numerical values
  - Some techniques work better/worse with features that have large numbers of possible values
    - For example, the feature **the word to the left** has a fairly large number of possible values

Naïve Bayes

- $\hat{s} = \underset{s \in S}{\arg\max}\ p(s|V)$, or $\underset{s \in S}{\arg\max} \frac{p(V|s)p(s)}{p(V)}$
- Where s is one of the senses S possible for a word w and V the input vector of feature values for w
- Assume features ***independent***, so probability of V is the product of probabilities of each feature, given s, so
- $$p(V|s) = \prod_{j=1}^{n} p(v_j|s)$$

  p(V) same for any ŝ

- Then $$\hat{s} = \underset{s \in S}{\arg\max}\ p(s) \prod_{j=1}^{n} p(v_j|s)$$

Naïve Bayes Test

- On a corpus of examples of uses of the word line, naïve Bayes achieved about 73% correct

- Good?

Decision Lists: another popular method

- A case statement….

Rule		Sense
fish within window	⇒	**bass**^{1}
striped bass	⇒	**bass**^{1}
guitar within window	⇒	**bass**^{2}
bass player	⇒	**bass**^{2}
piano within window	⇒	**bass**^{2}
tenor within window	⇒	**bass**^{2}
sea bass	⇒	**bass**^{1}
play/V *bass*	⇒	**bass**^{2}
river within window	⇒	**bass**^{1}
violin within window	⇒	**bass**^{2}
salmon within window	⇒	**bass**^{1}
on bass	⇒	**bass**^{2}
bass are	⇒	**bass**^{1}

Learning Decision Lists

- Restrict the lists to rules that test a single feature (1-decisionlist rules)
- Evaluate each possible test and rank them based on how well they work.
- Glue the top-N tests together and call that your decision list.

Yarowsky

- On a binary (homonymy) distinction used the following metric to rank the tests

$$\frac{P(\text{Sense}_1 \mid Feature)}{P(\text{Sense}_2 \mid Feature)}$$

- This gives about 95% on this test…

WSD Evaluations and baselines

- *In vivo* versus *in vitro* evaluation
- In vitro evaluation is most common now
  - Exact match **accuracy**
    - % of words tagged identically with manual sense tags
  - Usually evaluate using held-out data from same labeled corpus
    - Problems?
    - Why do we do it anyhow?
- Baselines
  - Most frequent sense

Most Frequent Sense

- Wordnet senses are ordered in frequency order
- So “most frequent sense” in wordnet = “take the first sense”

Freq	Synset	Gloss
338	plant[1], works, industrial plant	buildings for carrying on industrial labor
207	plant[2], flora, plant life	a living organism lacking the power of locomotion
2	plant[3]	something planted secretly for discovery by another
0	plant[4]	an actor situated in the audience whose acting is rehearsed but seems spontaneous to the audience

Ceiling

- Human inter-annotator agreement
  - Compare annotations of two humans
  - On same data
  - Given same tagging guidelines
- Human agreements on all-words corpora with Wordnet style senses
  - 75%-80%

Problems

- Given these general ML approaches, how many classifiers do I need to perform WSD robustly
  - One for each ambiguous word in the language
- How do you decide what set of tags/labels/senses to use for a given word?
  - Depends on the application

WordNet Bass

- Tagging with this set of senses is an impossibly hard task that's probably overkill for any realistic application

1. bass - (the lowest part of the musical range)
2. bass, bass part - (the lowest part in polyphonic music)
3. bass, basso - (an adult male singer with the lowest voice)
4. sea bass, bass - (flesh of lean-fleshed saltwater fish of the family Serranidae)
5. freshwater bass, bass - (any of various North American lean-fleshed freshwater fishes especially of the genus Micropterus)
6. bass, bass voice, basso - (the lowest adult male singing voice)
7. bass - (the member with the lowest range of a family of musical instruments)
8. bass -(nontechnical name for any of numerous edible marine and freshwater spiny-finned fishes)

Senseval History

- ACL-SIGLEX workshop (1997)
  - Yarowsky and Resnik paper
- SENSEVAL-I (1998)
  - Lexical Sample for English, French, and Italian
- SENSEVAL-II (Toulouse, 2001)
  - Lexical Sample and All Words
  - Organization: Kilkgarriff (Brighton)
- SENSEVAL-III (2004)
- SENSEVAL-IV -> SEMEVAL (2007)
- SEMEVAL (2010)
- SEMEVAL 2017: http://alt.qcri.org/semeval2017/index.php?id=tasks

SLIDE ADAPTED FROM CHRIS MANNING

WSD Performance

- Varies widely depending on how difficult the disambiguation task is
- Accuracies of over 90% are commonly reported on some of the classic, often fairly easy, WSD tasks (pike, star, interest)
- Senseval brought careful evaluation of difficult WSD (many senses, different POS)
- Senseval 1: more fine grained senses, wider range of types:
  - Overall: about 75% accuracy
  - Nouns: about 80% accuracy
  - Verbs: about 70% accuracy

What about word embeddings?

Summary

- Lexical Semantics
  - Homonymy, Polysemy, Synonymy
  - Thematic roles
- Computational resource for lexical semantics
  - WordNet
- Task
  - Word sense disambiguation

Where we left off

Where we left off

- BERT!

The Transformer architecture

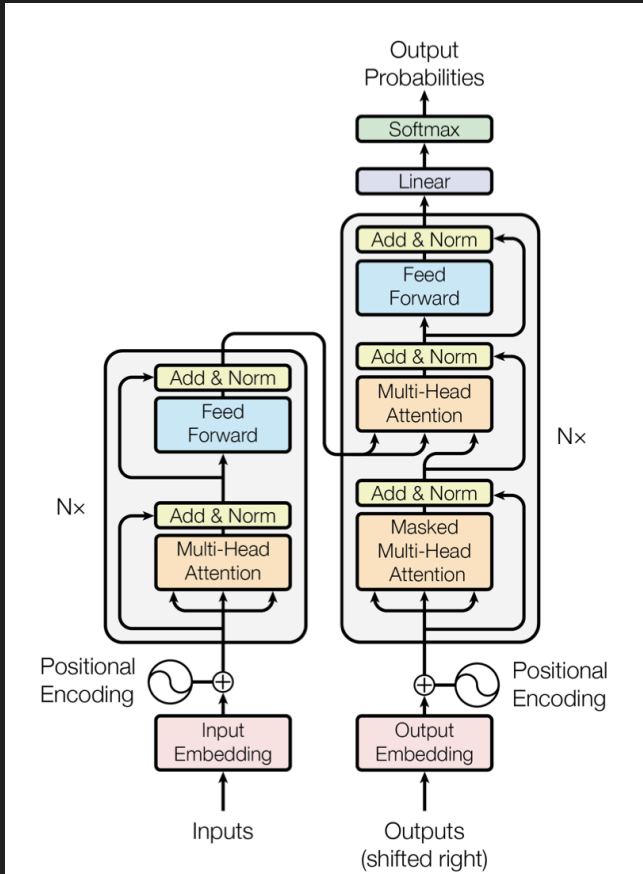


Figure 1: The Transformer - model architecture.

Figure: From Vaswani et al. [2017]

BERT, decomposed

That's a lot of moving parts!

But simpler than it looks. To begin with, BERT only uses the Transformer encoder. We'll look at this one piece at a time.

- Multi-head attention
- Position-wise feedforward layer
- "Add and Norm"
- Positional encoding

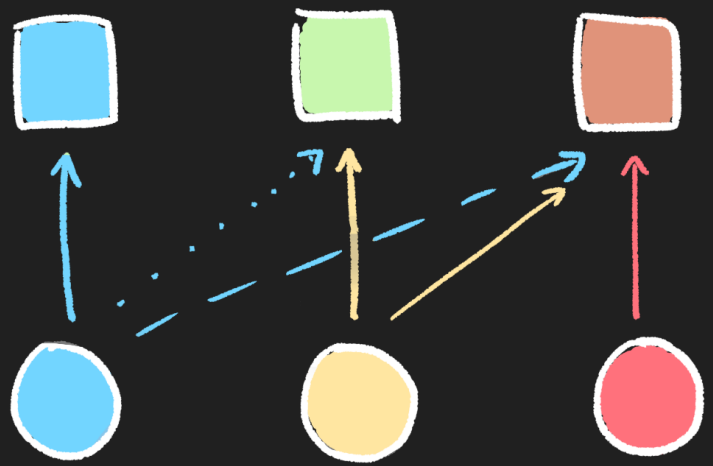
Multi-head(?) attention

- Multi-head: just several pieces of attention stacked together
- Scaled dot product attention: attention over *values*, weighted by similarity of *query* to *keys*

Attention, in overview

Idea: when examining an element of a sequence, useful to have information about other elements. So we 'mix' some of the other items into the representation at each position.

But some items are more important than others. So we do this with a *weighted average* of items in the sequence, using some weighting function.



Attention - setting

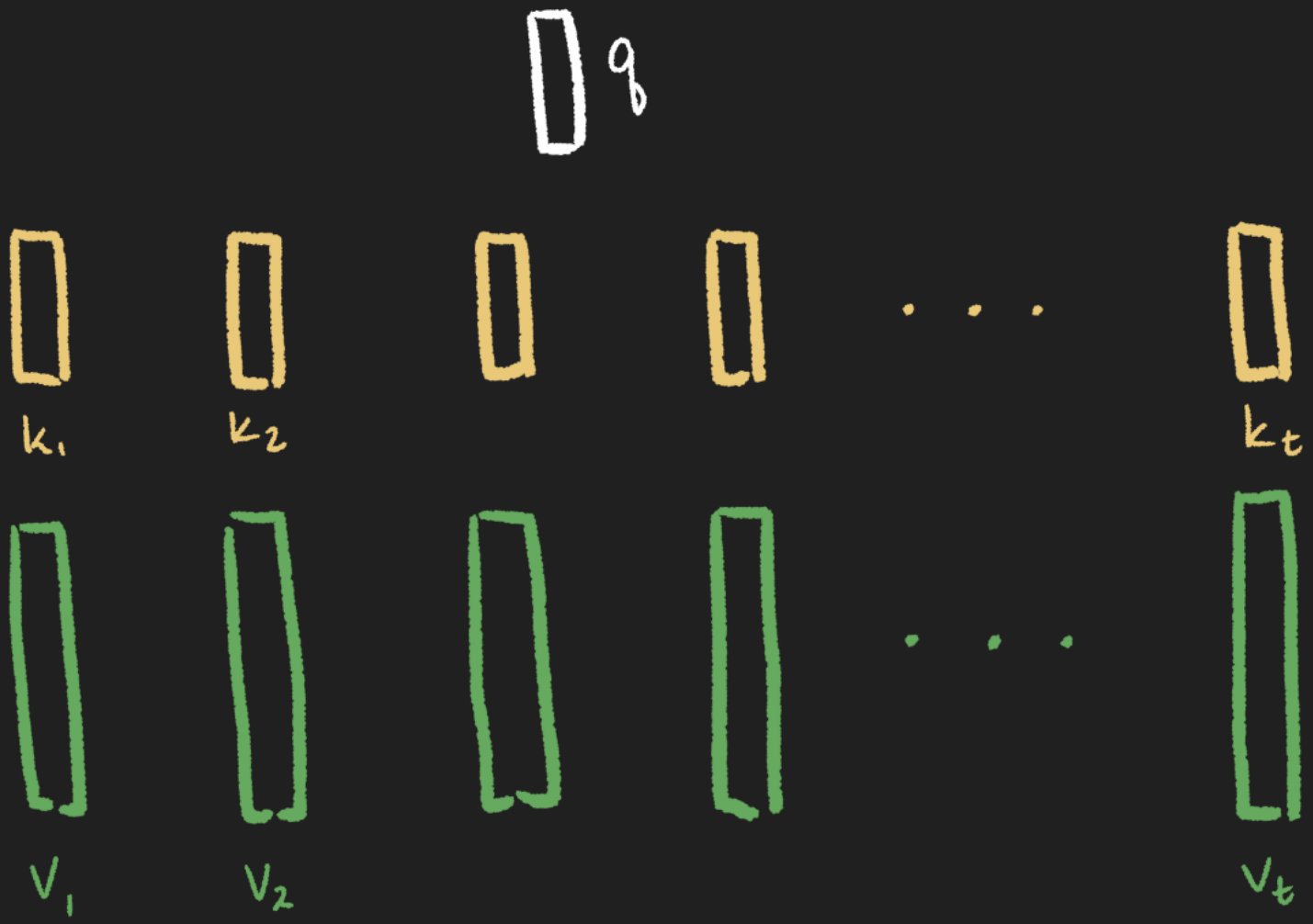
We have three sequences:

- A sequence of *keys* $\{k_1, ..., k_t\}$, $k_i \in \mathbb{R}^{d_k}$
- A sequence of corresponding *values* $\{v_1, ..., v_t\}$, $v_i \in \mathbb{R}^{d_v}$
- A sequence of *queries* $\{q_1, ..., q_t\}$, $q_i \in \mathbb{R}^{d_q}$, one for each position in our time series

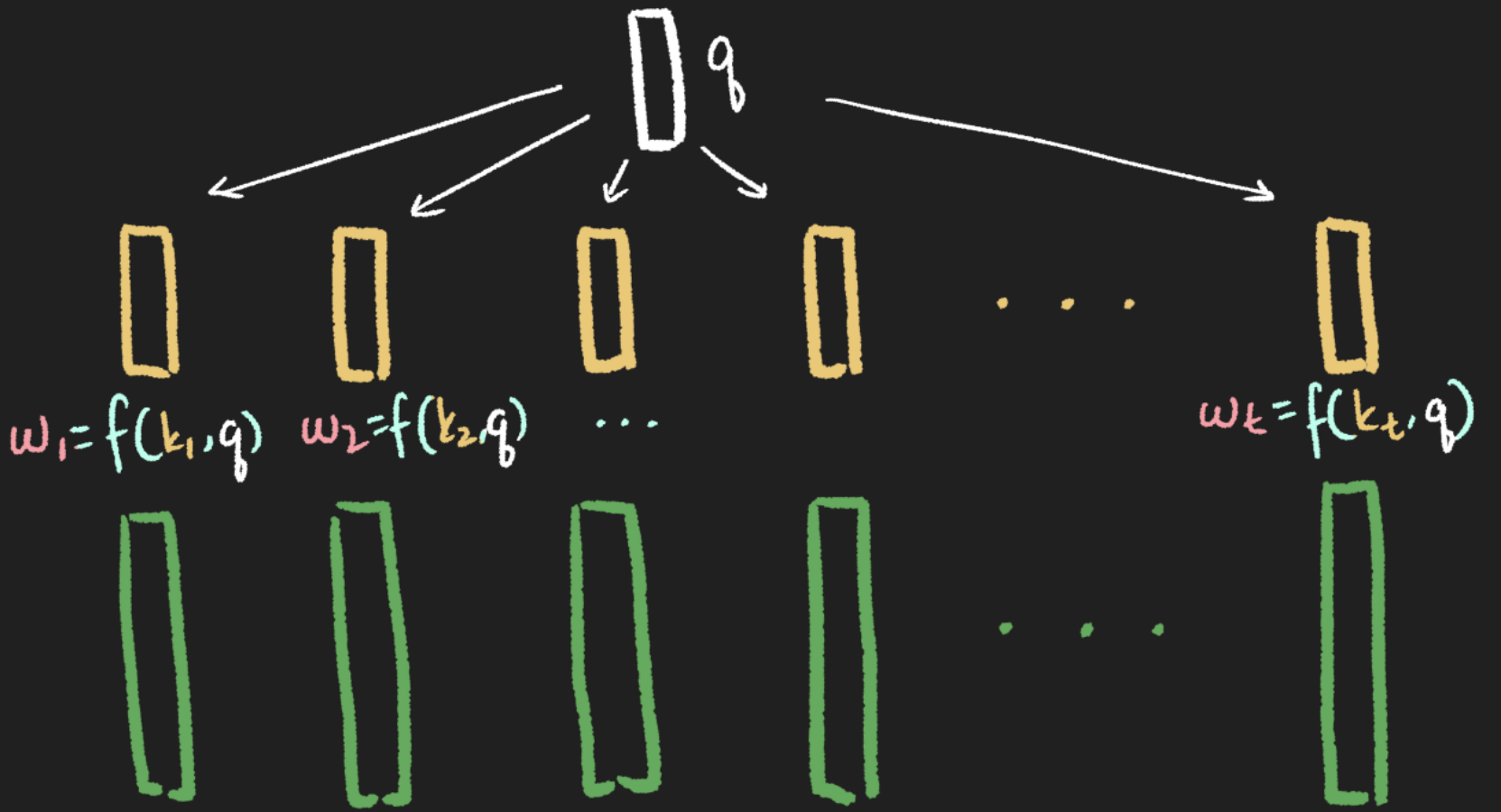
Then, using weighting function f, our output sequence will be $\{y_1, ..., y_t\}$, where

$$y_i = \sum_{i=1}^{t} f(k_i, q_i) v_i$$

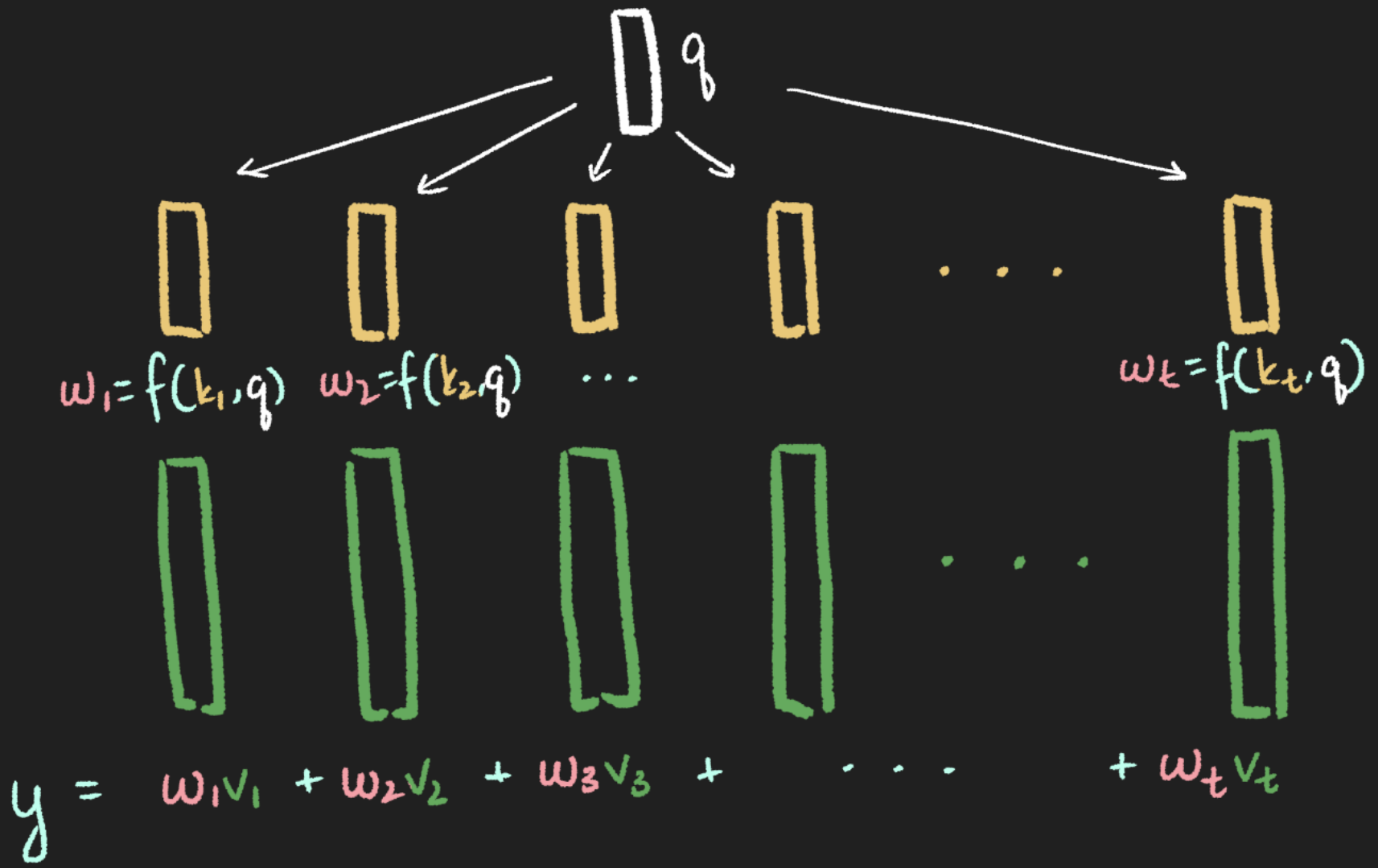
Attention - setting



Attention - setting



Attention - setting



Scaled dot product attention

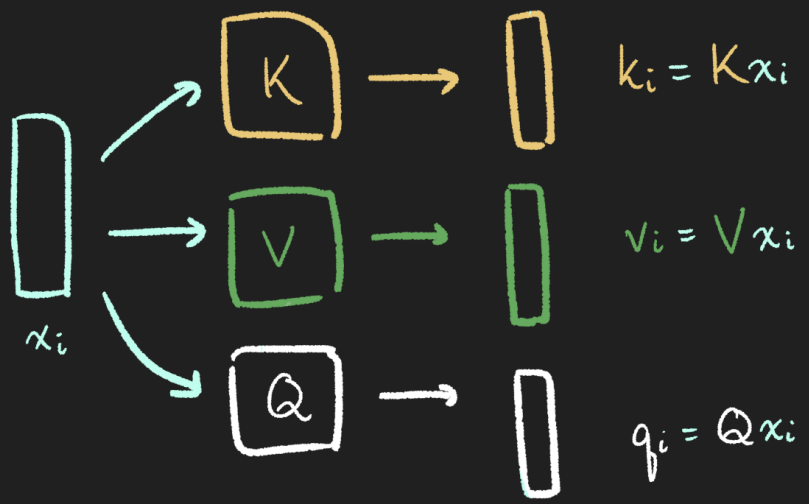
The kind of attention the Transformer architecture uses is pretty simple: the weighting function is just a normalized version of the dot product

$$y_i = \sum_{i=1}^{t} softmax\left(\frac{k_i \cdot q_i}{\sqrt{d_k}}\right) v_i.$$

But where do we get k_i, v_i and q_i? Simple - the Transformer encoder uses *self-attention*, so they're all (linear projections of) the same thing, the corresponding input vector x_i.

Self-attention and projections

What we're learning in the Transformer's attention layer are actually the projection matrices K, V and Q! (Like embedding weights, but with dense input.)



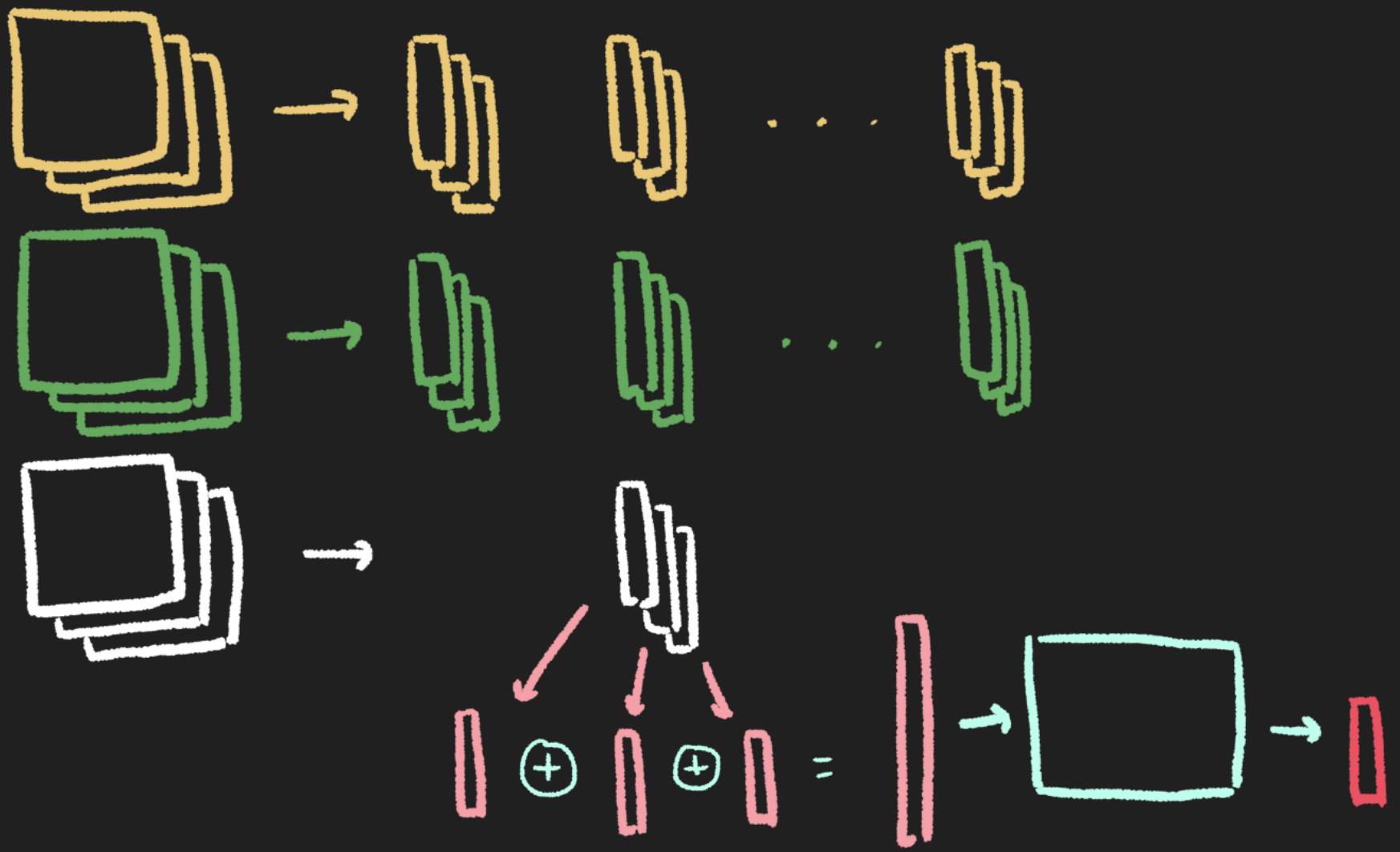
We compute keys and values with $k_i = Kx_i$, $v_i = Vx_i$, and the query $q_i = Qx_i$.

Multi-head attention

So what is this multi-head business?

Maybe one set of keys, values and queries isn't enough. So we'll actually learn h of these sets of projections in parallel. Then we'll concatenate them and take a linear projection back down to d_v. That's all.

Multi-head attention



Position-wise feedforward layer

Just a feedforward net (two layers and ReLU) applied to each sequence element individually.

What happens after each layer?

"Add and norm"

- Add the output $y = \sum_{i=1}^{t} softmax(\frac{k_i \cdot q}{\sqrt{d_k}})v_i$ to the input x
- Normalize $x + y$.

Easy.

Positional encoding

A collection of waveforms representing the position of each sequence element. This just helps keep track of what information came from where (since attention is a weighted sum, it naturally discards position information).

BERT

...doesn't modify the Transformer architecture at all.

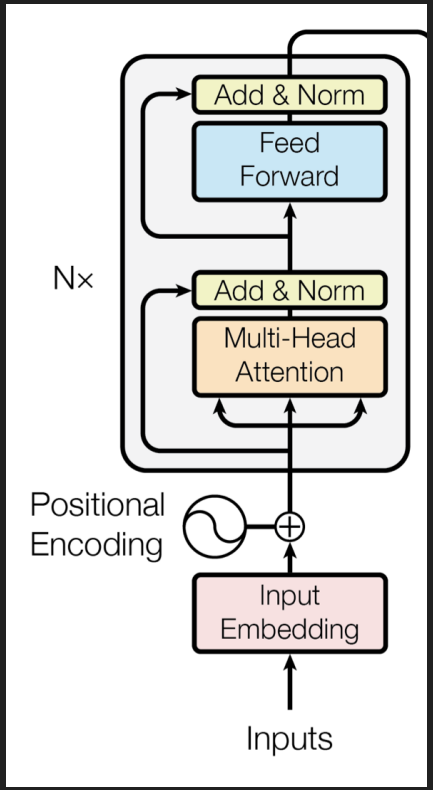


Figure: The Transformer encoder.

Training

Two training tasks:

- Masked word prediction
  - Sort of like a generalization of the regular LM task (predict the next word) - here, 'fill-in-the-blank'
  - "I knelt down to tie my ??? before walking out the door."
  - Mask out a token through all layers, and predict at the end
- Next sentence prediction
  - Given two sentences, does the second one really follow the first in the corpus, or are they unrelated?
  - Allows for easy generalization to specific tasks involving multiple sentences

Results?

New SOTA on multiple benchmarks.

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Figure: Results on GLUE, from Devlin et al. [2019] (highlights added)

- MNLI: multi-genre entailment prediction
- MSR paraphrase corpus

So what's actually going on here?

- Hewitt and Manning [2019] show that *entire syntax trees* can be recovered from BERT via simple linear projection
- Coenen et al. [2019] provide a visualization of BERT embeddings and elaborate on Hewitt and Manning's results
- Tenney et al. [2019] analyze the layers of BERT and find that it approximately follows the steps of a classical end-to-end NLP system

BERT and sense

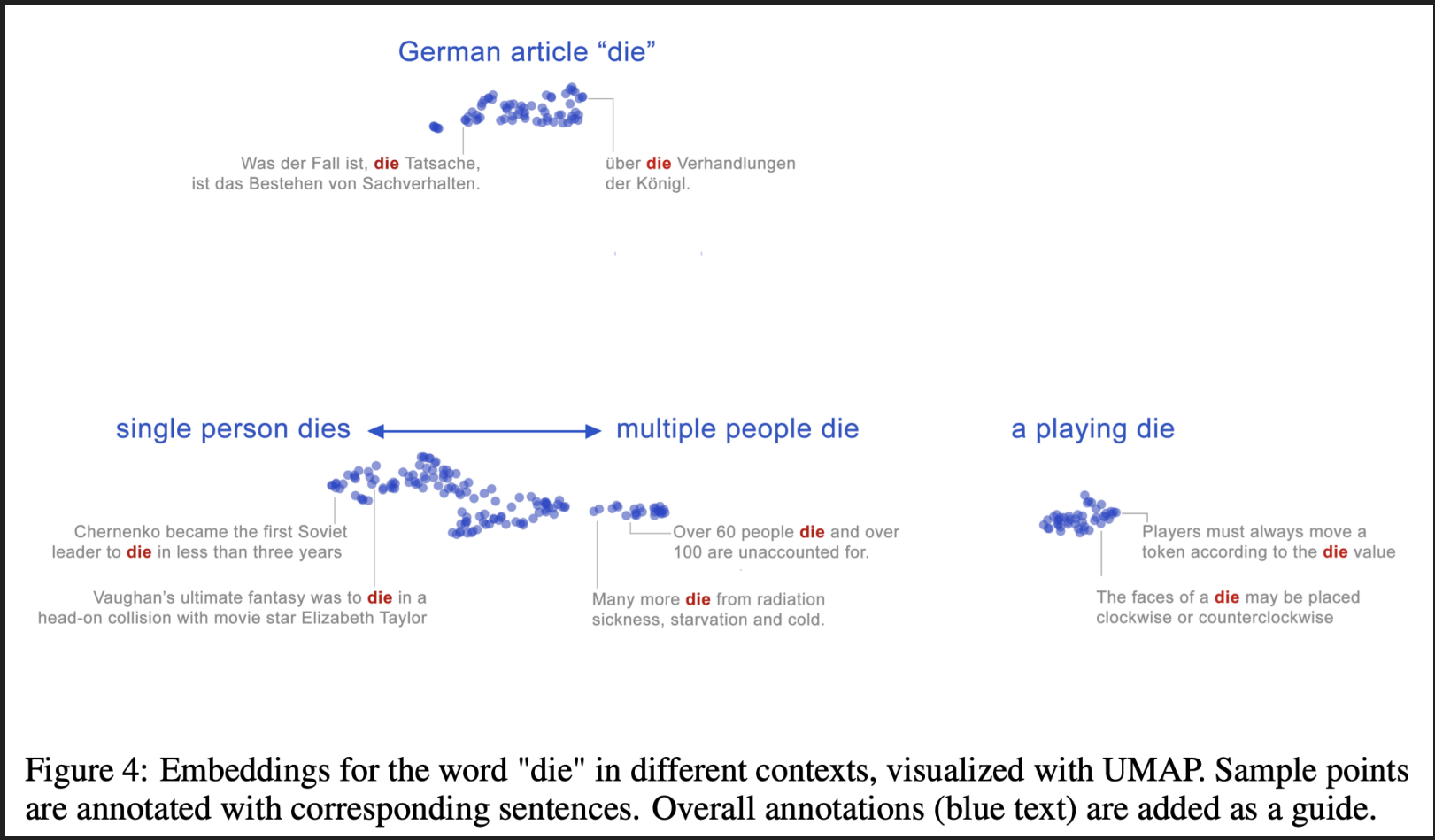


Figure 4: Embeddings for the word "die" in different contexts, visualized with UMAP. Sample points are annotated with corresponding sentences. Overall annotations (blue text) are added as a guide.

Figure: From Coenen et al. [2019]

BERT and sense disambiguation

What does this mean from a practical standpoint?

Even simple algorithms like nearest-neighbor classification on cluster centers do remarkably well.

BERT and WSD

Further work by Huang et al. [2019] shows even greater improvements.

- Idea: use *glosses* to discriminate
- Classify (sentence, gloss) pairs

	Dev	Test Datasets				Concatenation of Test Datasets				
System	SE07	SE2	SE3	SE13	SE15	Noun	Verb	Adj	Adv	**All**
MFS baseline	54.5	65.6	66.0	63.8	67.1	67.7	49.8	73.1	80.5	65.5
Lesk$_{ext+emb}$	56.7	63.0	63.7	66.2	64.6	70.0	51.1	51.7	80.6	64.2
Babelfy	51.6	67.0	63.5	66.4	70.3	68.9	50.7	73.2	79.8	66.4
IMS	61.3	70.9	69.3	65.3	69.5	70.5	55.8	75.6	82.9	68.9
IMS$_{+emb}$	62.6	72.2	70.4	65.9	71.5	71.9	56.6	75.9	84.7	70.1
Bi-LSTM	-	71.1	68.4	64.8	68.3	69.5	55.9	76.2	82.4	68.4
Bi-LSTM$_{+att.+LEX+POS}$	64.8	72.0	69.1	66.9	71.5	71.5	57.5	75.0	83.8	69.9
GAS$_{ext}$ (Linear)	-	72.4	70.1	67.1	72.1	71.9	58.1	76.4	84.7	70.4
GAS$_{ext}$ (Concatenation)	-	72.2	70.5	67.2	72.6	72.2	57.7	76.6	85.0	70.6
CAN^{s}	-	72.2	70.2	69.1	72.2	73.5	56.5	76.6	80.3	70.9
HCAN	-	72.8	70.3	68.5	72.8	72.7	58.2	77.4	84.1	71.1
BERT(Token-CLS)	61.1	69.7	69.4	65.8	69.5	72.0	57.8	73.5	84.4	68.6
GlossBERT(Sent-CLS)	69.2	76.5	73.4	75.1	79.5	79.1	65.4	79.3	84.8	75.8
GlossBERT(Token-CLS)	71.9	77.0	**75.4**	74.6	79.3	78.8	66.8	**79.9**	85.0	76.3
GlossBERT(Sent-CLS-WS)	**72.5**	**77.7**	75.2	**76.1**	**80.4**	**79.8**	**67.1**	79.6	**87.4**	**77.0**

Figure: Results from Huang et al. [2019].

In conclusion

Lots of work still to be done here!

Analysis of contextual embedding space is an exciting and still relatively new field, and there's lots to be discovered about how to use these representations

Thanks!

Questions?

References I

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda B. Viégas, and Martin Wattenberg. Visualizing and measuring the geometry of BERT. CoRR, abs/1906.02715, 2019. URL http://arxiv.org/abs/1906.02715.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

References II

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://www.aclweb.org/anthology/N19-1419.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. Glossbert: Bert for word sense disambiguation with gloss knowledge. arXiv preprint arXiv:1908.07245, 2019.

References III

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.