Word Vectors - FastText, ELMo

Natural Language Processing

(based on revision of Chris Manning Lectures)

Announcement

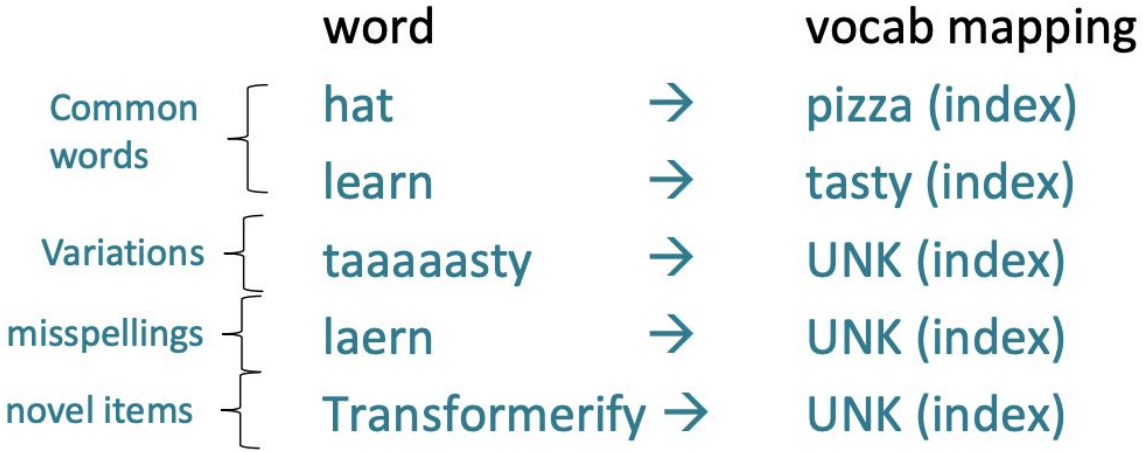
- TA announcements (if any)...

Suggested Readings

1. Contextual Word Representations: A Contextual Introduction
2. The Illustrated BERT, ELMo, and co.

Unknown (UNK)

All novel words seen at test time are mapped to a single UNK



Treating language as “words” does not make sense

Many languages exhibit complex **morphology**, or word structure

Example: **Swahili verbs can have hundreds on conjugations**, each encoding a wide variety of information. (Tense, mood, definiteness, negation, information, about the object, ++)

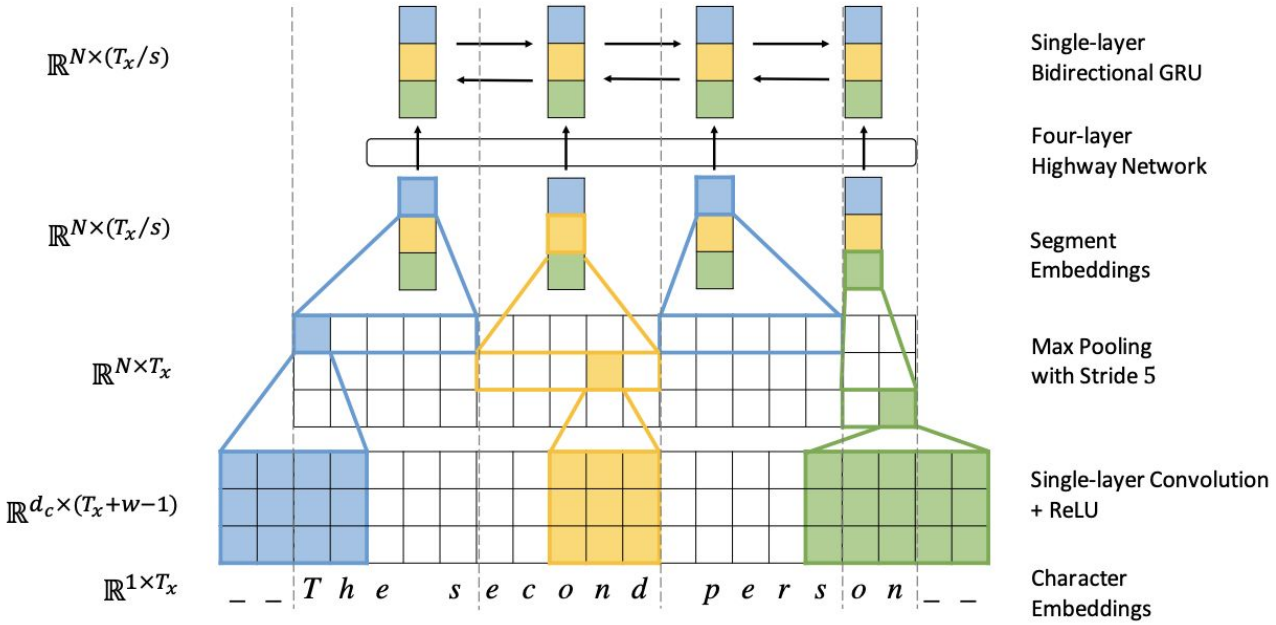
Thus researchers have started exploring character-based model, hybrid-based model, and subword-based models…..TLDR: it was more **effective**, especially on **solving out-of-vocabulary problems.**

Three approaches

- **Pure character level model**
  - Helps OOV problem
  - Treat each character separately
  - Time-consuming and not effective really....
- **Hybrid model**
  - Helps OOV problem
  - Treat only unknown words as characters
  - Still time consuming
- **Sub-word models**
  - Uses word segmentation algorithms
  - Treat some word as word, and some subwords as individual word
  - Best of both worlds (subwords + words)

Pure character level model

Pure character level model [Lee et al., ACL 2017]



Picture on the left is the encoder. The decoder is a char-level GRU.

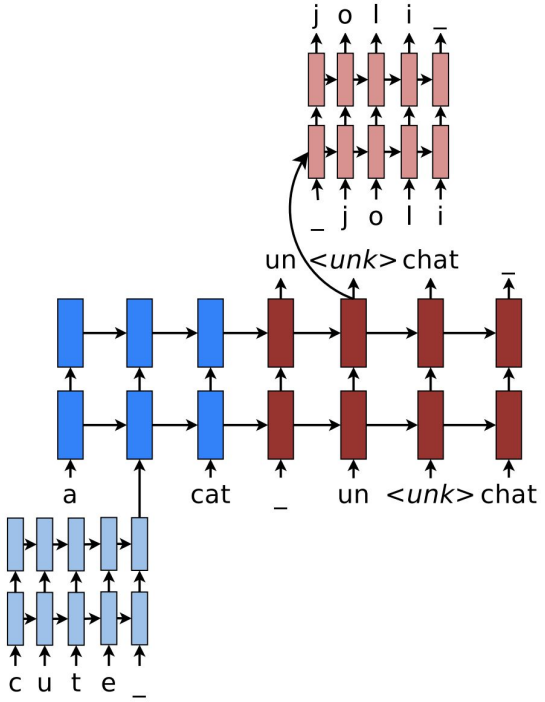
In Cs-En, they found that when the decoder is character-based (i.e., character-based beam search), BLEU increased by around 2, where compared to subword-based beam search....However, char-based encoder shows no improvements over subwords....

Fully Character-Level Neural Machine Translation without Explicit Segmentation, Lee et al., 2017, https://aclanthology.org/Q17-1026.pdf

Hybrid model

Hybrid Word-Character Models [Luong and Manning, ACL 2016]

- Translating “a cute cat” (English) into “un joli chat” (French)
- For rare tokens, here is the word “cute”, the rare token will be first inputted into some character-based LSTM to learn the representations. “_” is the boundary (similar to <START>)
- During teacher forcing, the target of this rare word will be set as “<unk>”
- For any “unk” symbol at that time step, the symbol will be further inputted into a LSTM to retrieve the word (i.e., char-level beam search). Here in French, cute is “joli”



Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models, Luong and Manning, 2016, https://arxiv.org/pdf/1604.00788.pdf

Hybrid Word-Character Models

	System	Vocab	Perplexity		BLEU	chrF$_3$
			w	c		
(a)	Best WMT'15, big data (Bojar and Tamchyna, 2015)	-	-	-	***18.8***	-
	Existing NMT					
(b)	RNNsearch + unk replace (Jean et al., 2015b)	200K	-	-	15.7	-
(c)	***Ensemble*** 4 models + unk replace (Jean et al., 2015b)	200K	-	-	18.3	-
	Our *word-based* NMT					
(d)	Base + attention + unk replace	50K	5.9	-	17.5	42.4
(e)	***Ensemble*** 4 models + unk replace	50K	-	-	18.4	43.9
	Our *character-based* NMT					
(f)	Base-512 (600-step backprop)	200	-	2.4	3.8	25.9
(g)	Base-512 + attention (600-step backprop)	200	-	1.6	17.5	***46.6***
(h)	Base-1024 + attention (300-step backprop)	200	-	1.9	15.7	41.1
	Our *hybrid* NMT					
(i)	Base + attention + same-path	10K	4.9	1.7	14.1	37.2
(j)	Base + attention + separate-path	10K	4.9	1.7	15.6	39.6
(k)	Base + attention + separate-path + 2-layer char	10K	4.7	1.6	***17.7***	44.1
(l)	Base + attention + separate-path + 2-layer char	50K	5.7	1.6	19.6	46.5
(m)	***Ensemble*** 4 models	50K	-	-	**20.7**	**47.5**

Hybrid Word-Character Models

1	source	The author ***Stephen Jay Gould*** died 20 years after ***diagnosis*** .
	human	Autor ***Stephen Jay Gould*** zemřel 20 let po ***diagnóze*** .
	word	Autor Stephen Jay <*unk*> zemřel 20 let po <*unk*> .
		Autor ***Stephen Jay Gould*** zemřel 20 let po **po** .
	char	Autor **Stepher Stepher** zemřel 20 let po ***diagnóze*** .
	hybrid	Autor <*unk*> <*unk*> <*unk*> zemřel 20 let po <*unk*>.
		Autor ***Stephen Jay Gould*** zemřel 20 let po ***diagnóze*** .
2	source	As the Reverend ***Martin Luther King Jr.*** said ***fifty years ago*** :
	human	Jak ***před padesáti lety*** řekl reverend ***Martin Luther King Jr*** . :
	word	Jak řekl reverend Martin <*unk*> King <*unk*> před padesáti lety :
		Jak řekl reverend ***Martin Luther King*** **řekl** před padesáti lety :
	char	Jako reverend ***Martin Luther*** **král říkal** před padesáti lety :
	hybrid	Jak před <*unk*> lety řekl <*unk*> Martin <*unk*> <*unk*> <*unk*> :
		Jak ***před padesáti lety*** řekl reverend ***Martin Luther King*** Jr. :
3	source	Her ***11-year-old*** daughter , ***Shani Bart*** , said it felt a " little bit ***weird*** " [..] back to school .
	human	Její ***jedenáctiletá*** dcera ***Shani Bartová*** prozradila , že " je to trochu ***zvláštní*** " [..] znova do školy .
	word	Její <*unk*> dcera <*unk*> <*unk*> řekla , že je to " trochu divné " , [..] vrací do školy .
		Její **11-year-old** dcera ***Shani*** , řekla , že je to " trochu divné " , [..] vrací do školy .
	char	Její ***jedenáctiletá*** dcera , ***Shani Bartová*** , říkala , že cítí trochu divně , [..] vrátila do školy .
	hybrid	Její <*unk*> dcera , <*unk*> <*unk*> , řekla , že cítí " trochu <*unk*> " , [..] vrátila do školy .
		Její ***jedenáctiletá*** dcera , **Graham** Bart , řekla , že cítí " trochu divný " , [..] vrátila do školy .

Table 4: **Sample translations on newstest2015** – for each example, we show the *source*, *human* translation, and translations of the following NMT systems: *word* model *(d)*, *char* model *(g)*, and *hybrid* model *(k)*. We show the translations before replacing <*unk*> tokens (if any) for the word-based and hybrid models. The following formats are used to highlight ***correct***, **wrong**, and close translation segments.

In the first example, the hybrid model translates perfectly. The word-based model fails to translate “diagnosis” because the second <unk> was incorrectly aligned to the word “after”.

The character-based model, on the other hand, makes a mistake in translating names.

Subword-based models

How do we “segment” words, so they are subwords?

Byte Pair Encoding is a word segmentation algorithm

- Start with a vocabulary of characters
- Most frequent n-gram pairs -> a new gram

Dictionary

5	l o w
2	l o w e r
6	n e w e s t
3	w i d e s t

Vocabulary

l, o, w, e, r, n, w, s, t, i, d

Start with all characters in vocab

How do we “segment” words, so they are subwords?

Byte Pair Encoding is a word segmentation algorithm

- Start with a vocabulary of characters
- Most frequent n-gram pairs -> a new gram

Dictionary

5	l o w
2	l o w e r
6	n e w **e s** t
3	w i d **e s** t

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, **es**

Add a pair (e, s) with freq 9

How do we “segment” words, so they are subwords?

Byte Pair Encoding is a word segmentation algorithm

- Start with a vocabulary of characters
- Most frequent n-gram pairs -> a new gram

Dictionary

5	l o w
2	l o w e r
6	n e w **e s t**
3	w i d **e s t**

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, **est**

Add a pair (es, t) with freq 9

How do we “segment” words, so they are subwords?

Byte Pair Encoding is a word segmentation algorithm

- Start with a vocabulary of characters
- Most frequent n-gram pairs -> a new gram

Dictionary

5	**l o** w
2	**l o** w e r
6	n e w e s t
3	w i d e s t

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, est, **lo**

Add a pair (l, o) with freq 7

How do we “segment” words, so they are subwords?

- Stop when we reach our desired vocabulary size
- **Google NMT** uses a variant of this called “**Word Piece**” model
  - Very similar, i.e., instead of using n-gram frequencies, they used n-gram that maximally reduces **perplexity**
- **BERT** uses a variant of the **wordpiece** model
  - (Relatively) common words are in the vocabulary
    - *at, fridge, 1910s*
  - Other words are built from wordpieces:
    - Hypatia = h ##yp ##ati ##a
      - Non-word-initial units are prefixed with ## (different system uses slightly different ways to annotate)
- Now, subwords are used prominently, and mostly in a pre-trained fashion (cover more later in L11)

FastText

FastText [Bojanowski et al., ACL 2017]

- Same author as Word2vec
- Train similarly like Word2vec with skip-gram architecture
- But represent words as **n-grams**.
- For example, let say n=3
  - where = <wh, whe, her, ere, re>, <where>
  - Note that "<" and ">" are considered as characters
- Represent word as **sum** of these representations

		sg	cbow	sisg-	sisg
AR	WS353	51	52	54	**55**
DE	GUR350	61	62	64	**70**
	GUR65	78	78	**81**	**81**
	ZG222	35	38	41	**44**
EN	RW	43	43	46	**47**
	WS353	72	**73**	71	71
ES	WS353	57	58	58	**59**
FR	RG65	70	69	**75**	**75**
RO	WS353	48	52	51	**54**
RU	HJ	59	60	60	**66**

Table 1: Correlation between human judgement and similarity scores on word similarity datasets. We train both our model and the `word2vec` baseline on normalized Wikipedia dumps. Evaluation datasets contain words that are not part of the training set, so we represent them using null vectors (`sisg-`). With our model, we also compute vectors for unseen words by summing the n-gram vectors (`sisg`).

Enriching Word Vectors with Subword Information, Bojanowski et al., 2017, https://arxiv.org/pdf/1607.04606.pdf

ELMo

Problems of Word Embeddings

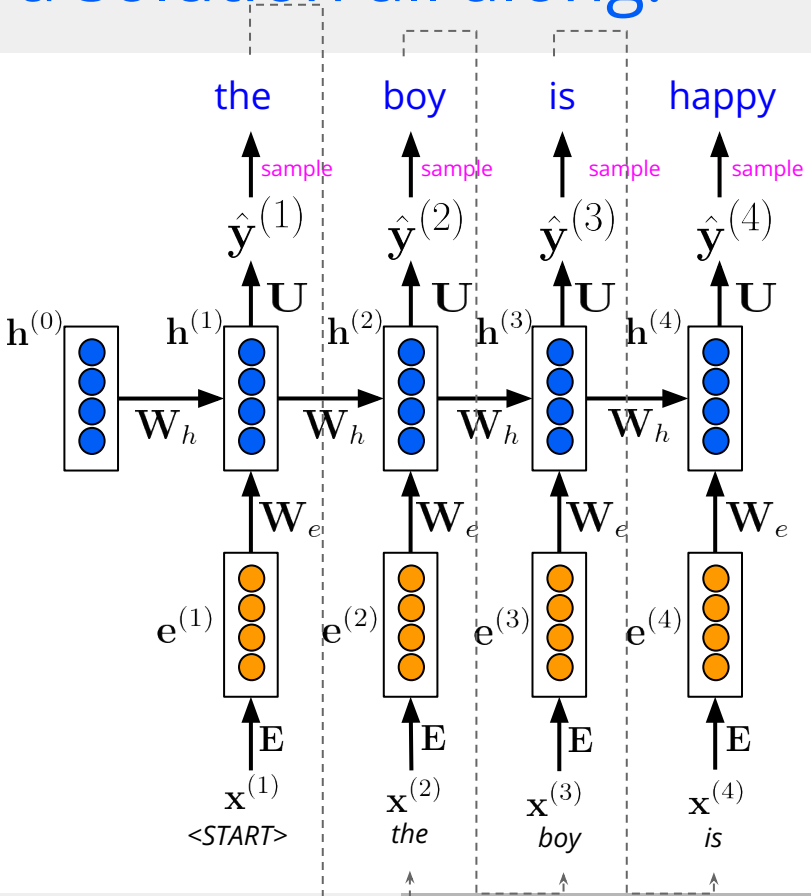
- Originally, we basically had one representation of words
  - The word vectors that we learned about at the beginning
    - Word2vec, GloVe, fastText

- These have two problems:
  - Always the same representation for a **word type** regardless of the context in which a **word token** occurs
  - We just have one representation for a word, but words have different **aspects**, including semantics, syntactic behavior, and register/connotations

We actually already have a solution all along!

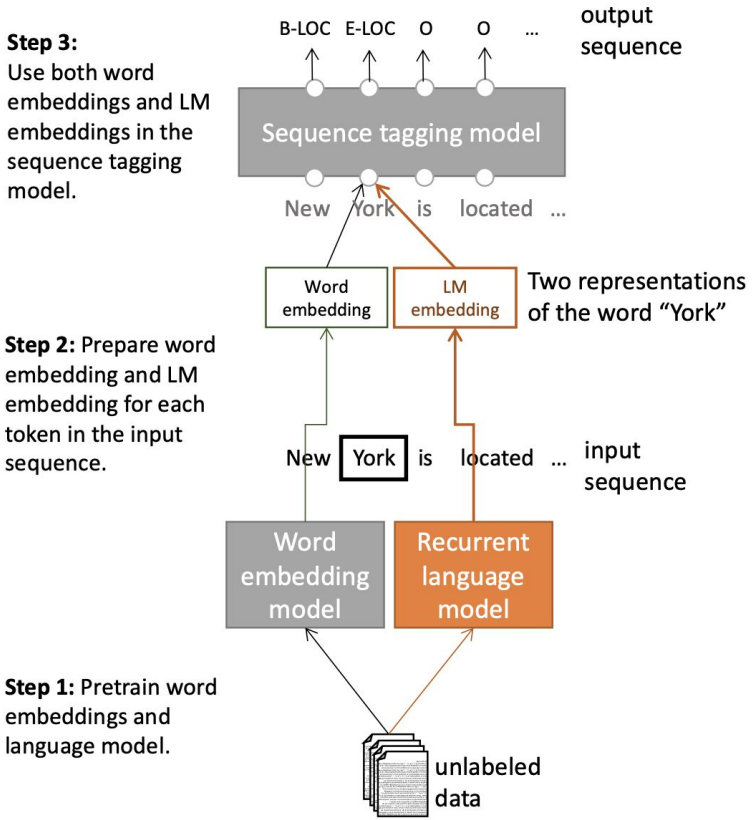
These **hidden states** actually embed contextual information!

Why not we train LM on a very large corpus, and take these hidden states for other purposes, e.g., classification?

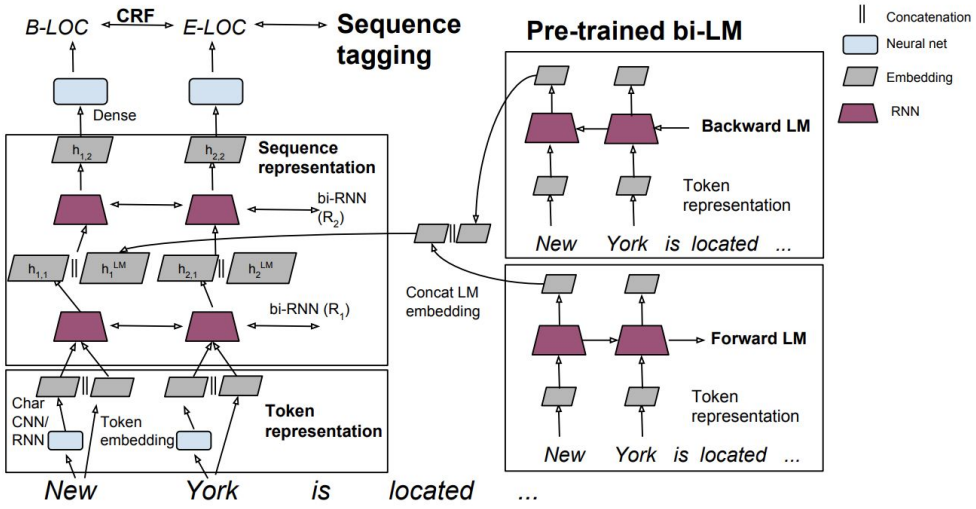


Tag LM [Peters et al. ACL 2017]

Semi-supervised sequence tagging with bidirectional language models, Peters et al. 2017, https://arxiv.org/pdf/1705.00108.pdf



Tag LM



Model	$F_1 \pm$ std
Chiu and Nichols (2016)	90.91 ± 0.20
Lample et al. (2016)	90.94
Ma and Hovy (2016)	91.37
Our baseline without LM	90.87 ± 0.13
TagLM	$\mathbf{91.93 \pm 0.19}$

Table 1: Test set F_1 comparison on CoNLL 2003 NER task, using only CoNLL 2003 data and unlabeled text.

ELMo: Embeddings from Language Models [Peters et al. ACL 2018]

- Learn a deep Bi-NLM and use all its layers in prediction
  - Use 2 biLSTM layers
    - First lower layer aims for lower-level syntax (POS, tagging, NER)
    - Higher layer is for higher level semantics (sentiment, semantic role, etc.)
- First run biLM to get representations for each word
  - Took **learned weighted average** of the 2 hidden layers
- Then let (whatever) end-task model use them
  - Freeze weights of ELMo for purposes of supervised model
  - Concatenate ELMo weights into task-specific model
    - Concatenating into intermediate layers just like TagLM
    - Can provide ELMo representations again, when producing outputs, e.g., in a question answering system

Deep contextualized word representations, Peters et al., 2018, https://arxiv.org/pdf/1802.05365.pdf

ELMo: Embeddings from Language Models

TASK	PREVIOUS SOTA		OUR BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

Summary

- **Char-based models** (similar to hybrid models) seem to be able to address out-of-vocabulary problem, but they are very slow to run...thus did not really receive much attention after 2017...
- Word embeddings such as **Word2vec** and **GloVe** suffers from **out-of-vocabulary** problem
  - **FastText** kinda addresses it
- Using the **hidden states as contextual vectors** - the core idea behind ELMo
  - Later on Analysis of Model's Inner Workings, scientists actually really find amazing things inside each hidden states of a huge pretrained model!
- **ELMo** and **ULMfit** are among many **inspirations** behind BERT and other pretrained models
- Later on, you will also learn **pretrained models like BERT**
  - Why not just pretrained the whole model on a **really large corpus**
  - Then reuse the top layer hidden states for any tasks. It works really well and is currently (2021) the SOTA (state of the art)
  - Word2vec, GloVe, FastText, and ELMo are NO longer being used once BERT has been introduced ....