NLP Paper Reading

Chang Liu

Joint Inference of Entities, Relations, and Coreference

UMass Amherst Singh et.

Target & Contribution

- Currently, ***joint inference*** approach to avoid cascading error of multiple NLP tasks is modest, with limits to modeling only two tasks at a time
- In this paper, they focus on **three** tasks of automated extraction pipeline: entity tagging, relation extraction and coreference
- They propose a single **joint graphical model** that allows the flow of uncertainty across the task boundaries
- They present a novel **extension to belief propagation(BP)** that sparsifies the domains of variables during inference.

Background

- Isolated models works well in separate task, but not in pipeline
- uni-directional flow, suffer from cascading error
- Previous: 1) only two tasks considered 2) coreference is not considered
- Now: 1) joint probabilistic graphical model 2) modification to BP that facilitates the efficient inference

Figure 1: **Information Extraction:** 3 mentions labeled with entity types (red), relations (green), and coreference (blue links).

Isolated Model

- Entity Tagging
  - mention **m_i**, predict label **t_i** in predefined ***L***
  - classification, maximum entropy model
- Relation Extraction
  - labels each entity mention pair with its its relation in a sentence
  - independently label each entity pair with its type
- Coreference
  - link mention that refers to real-word entity
  - classify pairs of mentions as coreferent or not

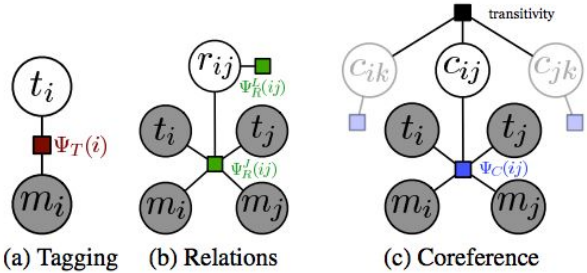
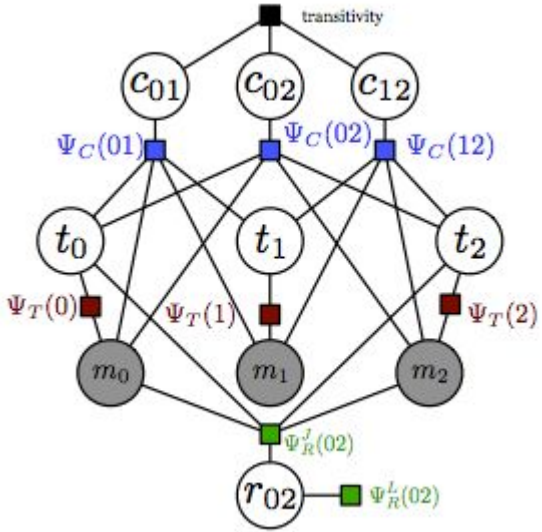


Figure 2: **Individual Classification Models:** where the observed/fixed variables are shaded in grey. For brevity, we have only included (i, j) in the factor labels, and have omitted t_i, t_j, m_i and m_j.

Joint Model

- Motivation
  - relation extraction and coreference models both use ***fixed*** entity tags, while not possible for entity tagging to benefit from this decision, resulting in uni-directional flow of information out of the entity tagging model
  - define the model that directly represents the dependence between these three tasks by modeling the joint distribution over the three tasks.
    - combine all the variables and factors into a single graphical model
    - do NOT fix the entity tags to be



Problem? Architecture

- With such **small** set of mention, the joint model is quite complex and dense
- Include T, R, C
- when training, this factor can capture the bi-directional information that flows through the task
- coreference and relation is not directly connected, as dependence is weak, but not independently. (tagging and relation is connected)

$$p(\mathbf{t},\mathbf{r},\mathbf{c}|\mathbf{m}) \propto \prod_{t_i \in \mathbf{t}} \Psi_T(m_i, t_i) \prod_{c_{ij} \in \mathbf{c}} \Psi_C(c_{ij}, m_i, m_j, t_i, t_j) \prod_{r_{ij} \in \mathbf{r}} \Psi_R^L(m_i, m_j, r_{ij}) \Psi_R^J(r_{ij}, m_i, m_j, t_i, t_j)$$

Learning & Inference

- **Piecewise learning** for joint model
  - common approach not working here(why?)
    - inference in the inner loop(NP-hard)
    - likelihood is defined over all tasks, cannot optimize balancing all the tasks
    - number of parameters so huge
  - How they do?
    - treat each factor as an independent piece, learning distribution given neighboring variables
    - entity tagging factors are incorporated during piecewise training of relation and coreference as fixed incoming belief(faster convergence)
    - only include the support features(i.e feature appears at least once in the training data)

Learning & Inference(const.)

- Sparsity for Efficient Inference
  - Why?
    - incredibly loopy, due to large number of factors
    - cannot use the belief propagation(BP) directly
  - Alternatives?
    - MCMC-based sampling (local minima, customized proposal function)
    - Integer linear program(ILP)
  - Motivation
    - detecting the low-entropy marginals in earlier phases and fixing to high-probability values provide benefits for BP
  - How?
    - Adapt algorithms for inference in the model
      - examine the marginals of all variables after every iteration
      - When it goes beyond a threshold s, set it to the ***fixed*** maximum probability

Experiment & Result

Data	#Mentions	#Coreference	#Relation
Train	15,640	637,160	82,479
Dev	5,545	244,461	34,057
Test	6,598	342,942	38,270

Table 1: Number of variables in the various folds.

Model	Accuracy	Error Red.
Isolated Model	80.23	-
Joint w/ Coreference	81.24	5.1
Joint w/ Relations	81.77	7.8
Complete Joint	**82.69**	12.4

Table 2: **Entity Tagging:** Results for various models.

Model	Prec	Rec	F1
Pipeline (w/ Tagging)	53.22	54.92	54.05
Joint w/ Tagging	54.93	54.02	54.47
Complete Joint	56.06	54.74	**55.39**

Table 3: **Relation Extraction:** Comparison using the F-measure.

Model	MUC	Pairwise	B^3
Pipeline (w/ Tagging)	**73.81**	53.94	76.34
Joint w/ Tagging	71.09	57.59	78.06
Complete Joint	73.00	**58.39**	**78.50**

Table 4: **Coreference Resolution:** MUC metric has been provided for comparison to exiting work; it is much less informative compared to Pairwise and B^3 since simple baselines attain high scores.

Conclusion

- introduce a novel, fully-joint model for three crucial information extraction tasks.
- introduce extension to BP that sparsifies variables during inference, which facilitate efficient inference
- combination of a joint model, with an accompanying inference technique, results in improvements to all three tasks
- improved representation of multiple task in the same model is beneficial to all the tasks

Evaluation of Word Vector Representations by Subspace Alignment

CMU Tsvetkov et.

Target & Contribution

- Currently, Most common **intrinsic** evaluations of vector quality measure correlation with similarity judgments
- these often **correlate poorly** with how well the learned representations perform as features in downstream evaluation tasks
- They present **QVEC**
  - computationally *inexpensive* intrinsic **evaluation measure**
  - measures the quality of **word embeddings** based on **alignment** to a matrix of features extracted from **manually crafted** lexical resources
  - **strong correlation** of the vectors in the downstream semantic evaluation tasks

Introduction

- **vector space word representation**: can be derived from large **unannotated** corpora, as a source of downstream NLP tasks
- No standard scheme for evaluating the quality of word vectors
  - quality is judged by its utility in downstream NLP tasks
  - due to word vector's criticism
    - linguistically opaque in a sense that it's not clear how to interpret individual vector dimension
    - not clear how to score a non-interpretable representation
- They propose simple, intrinsic evaluation for word vectors
  - based on component-wise correlations with **manually constructed** linguistic word vectors
  - their measure favors **recall**(instead of precision), captures intuition that **meaningless dimensions** are less harmful than **important dimensions that are missing**
  - sum of correlation of the **aligned dimensions** is the evaluation score

Linguistic Dimension Word Vectors

- The crux of evaluation lies in **quantifying** the **similarity** between a **distributional** word vector model and a (**gold-standard**) linguistic resource capturing human knowledge
- exploit an existing semantic resource - **SemCor**, construct a set of linguistic word vectors
- SemCor is a **WordNet**-annotated corpus that captures, among others, supersense(26 for nouns and 15 for verbs) annotations of WordNet's 13,174 noun lemmas and 5,686 verb lemmas at least once

WORD	NN.ANIMAL	NN.FOOD	...	VB.MOTION
fish	0.68	0.16	...	0.00
duck	0.31	0.00	...	0.69
chicken	0.33	0.67	...	0.00

Table 1: Oracle linguistic word vectors, constructed from a linguistic resource containing semantic annotations.

Word Vector Evaluation Model

- Align dimensions of distributed word vectors to dimension in the linguistic vectors
- By projection, we can obtains annotation of dimensions in the distributional word vectors
- distribute: **D** X **N**
- linguistic: **P** X **N**
- **1-n** alignment

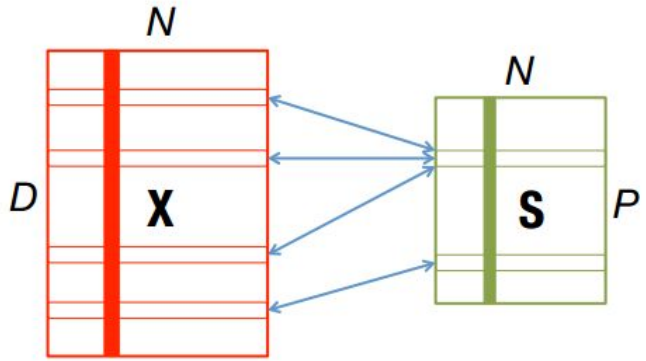


Figure 1: The filled vertical vectors represent the word vector in the word vector matrix **X** and the linguistic property matrix **S**. The horizontal hollow vectors represent the "distributional dimension vector" in **X** and "linguistic dimension vector" in **S**. The arrows show mapping between distributional and linguistic vector dimensions.

Word Vector Evaluation Model(const.)

- Model

Let $\mathbf{A} \in \{0,1\}^{D \times P}$ be a matrix of alignments such that $a_{ij} = 1$ iff $\mathbf{x}_i$ is aligned to $\mathbf{s}_j$, otherwise $a_{ij} = 0$. If $r(\mathbf{x}_i, \mathbf{s}_j)$ is the Pearson's correlation between vectors $\mathbf{x}_i$ and $\mathbf{s}_j$, then our objective is defined as:

$$\text{QVEC} = \max_{\mathbf{A} | \sum_j a_{ij} \leq 1} \sum_{i=1}^{L} \sum_{j=1}^{P} r(\mathbf{x}_i, \mathbf{s}_j) \times a_{ij} \quad (1)$$

- Limitation
  - dimension in linguistic matrix S may not capture every possible linguistic property
  - **recall-oriented** measure
    - highly correlated alignments provide evaluation
    - missing information doesn't significantly affect the score

Experiment & Result

- To test **QVEC**, select many word vector models
  - **CBOW and Skip-Gram(SG)**
    - each word's Huffman code
    - log-linear classifier
  - **CWindow and Structured SKip-Gram(SSG)**
    - syntactic modification to WORD2VEC
  - **CBOW with Attention(Attention)**
    - Word2VEC CBOW MODEL
  - **GloVe.**
  - **Latent Semantic Analysis(LSA)**
  - **GloVe+WN**
  - **GloVe+PPDB**
  - **LSA+WN**
  - **LSA+PPDB**

Model	QVEC	Senti
CBOW	40.3	90.0
SG	35.9	80.5
CWindow	28.1	76.2
SSG	40.5	81.2
Attention	40.8	80.1
GloVe	34.4	79.4
GloVe+WN	42.1	79.6
GloVe+PPDB	39.2	79.7
LSA	19.7	76.9
LSA+WN	29.4	77.5
LSA+PPDB	28.4	77.3
Correlation (r)	**0.87**	

Table 2: Intrinsic (QVEC) and extrinsic scores of the 300-dimensional vectors trained using different word vector models and evaluated on the Senti task. Pearson's correlation between the intrinsic and extrinsic scores is $r = 0.87$.

Experiment & Result(const.)

- Compare **QVEC** to six standard extrinsic semantic tasks
  - Word similarity
    - benchmark(**WS-353, MEN, SimLex-999**)
    - cosine similarity between ranking by model and human rankings
  - Text Classification
    - **20NG** dataset
    - **Senti** analysis
    - **Metaphor** detection

	20NG	Metaphor	Senti
WS-353	0.55	0.25	0.46
MEN	**0.76**	0.49	0.55
SimLex	0.56	0.44	0.51
QVEC	0.74	**0.75**	**0.88**

Table 4: Pearson's correlations between word similarity/QVEC scores and the downstream text classification tasks.

Result(const.)

	WS-353	MEN	SimLex	20NG	Metaphor	Senti
r	0.34	0.63	0.68	0.74	0.75	0.88

Table 3: Pearson's correlations between QVEC scores of the 300-dimensional vectors trained using different word vector models and the scores of the downstream tasks on the same vectors.

	50	100	200	300	500	1000
ρ(QVEC, Senti)	0.32	0.57	0.73	0.78	0.72	0.60
ρ(QVEC, All)	0.66	0.59	0.63	0.65	0.62	0.59

Table 5: Spearman's rank-order correlation between the QVEC ranking of the word vector models and the ranking produced by (1) the Senti task, or (2) the aggregated ranking of all tasks (All). We rank separately models of vectors of different dimensionality (table columns).

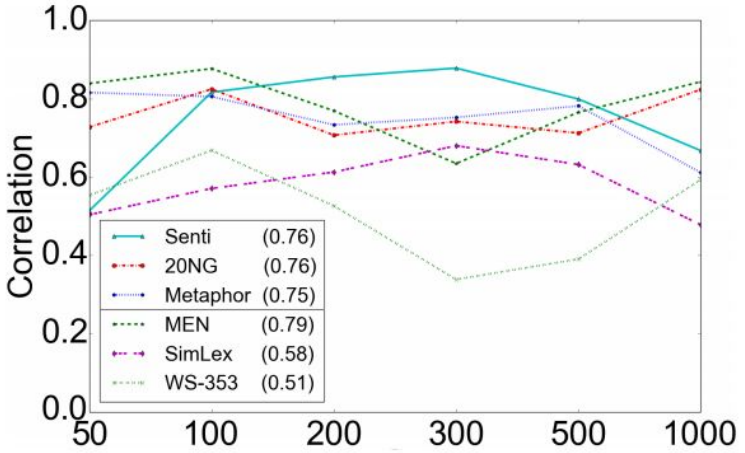


Figure 2: Pearson's correlation between QVEC scores and the semantic benchmarks across word vector models on vectors of different dimensionality. The scores at dimension 300 correspond to the results shown in table 3. The scores in the legend show average correlation across dimensions.

Summary

- To summarize, we observe high positive correlation between **QVEC** and the downstream tasks, consistent across the tasks and across different modelswith vectors of different dimensionalities
- We propose a method for intrinsic evaluation of word vectors which shows strong relationship—both linear and monotonic—with the scores/rankings produced by the downstream tasks.