Data-Driven Dialogue State Tracking using (Specialised) Word Embeddings

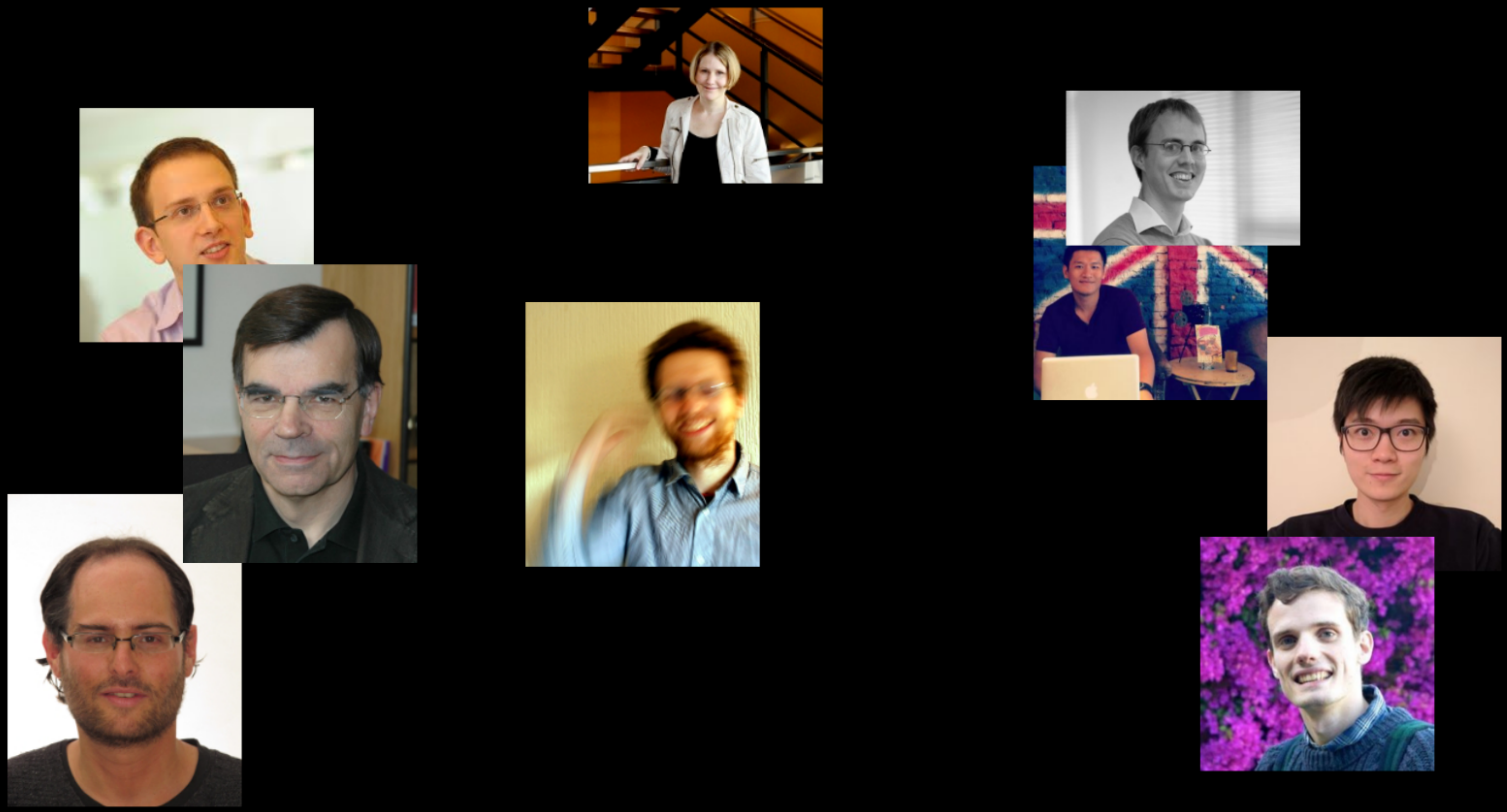
Ivan Vulić

University of Cambridge and **PolyAI**

Conversational Intelligence Summer School

Moscow; July 5, 2018

Joint Work with...



...plus efforts of many other researchers...

Talk Overview

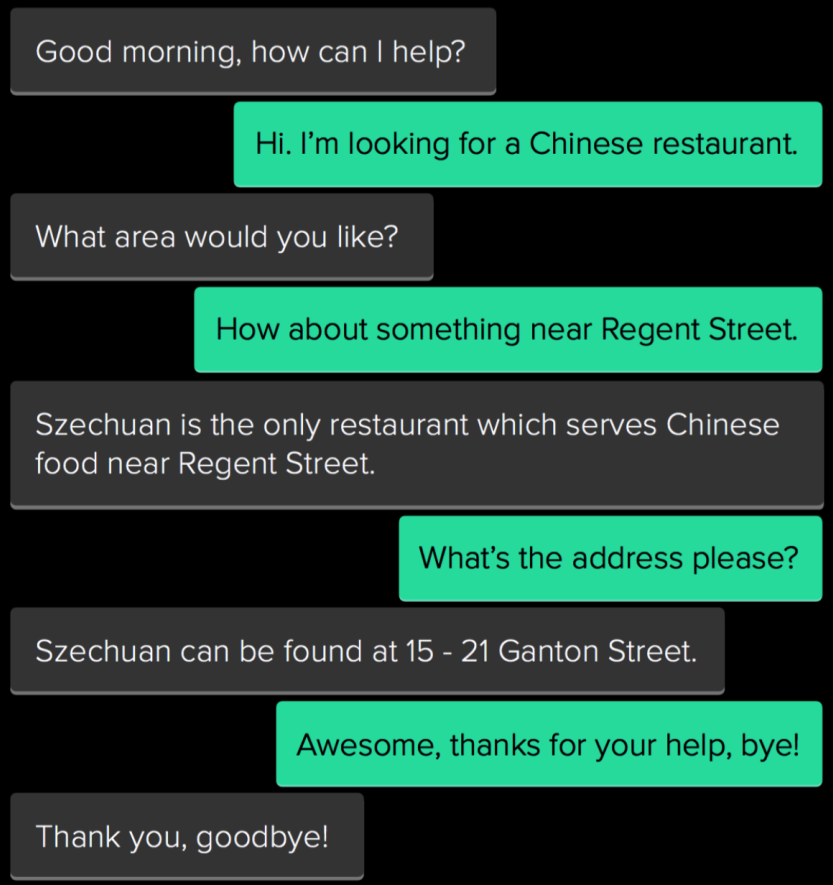
Dialogue State Tracking: Real-World Language Understanding Task

- Dialogue State Tracking: Problem Definition
- Semantic Specialisation: Word Vectors to the Rescue?
- Attract-Repel: State-of-the-Art Specialisation Method
- Neural Belief Tracker: Data-Driven Dialogue State Tracking
- Bootstrapping Models for Lower-Resource Languages
- Recent Improvements to the Original NBT Framework

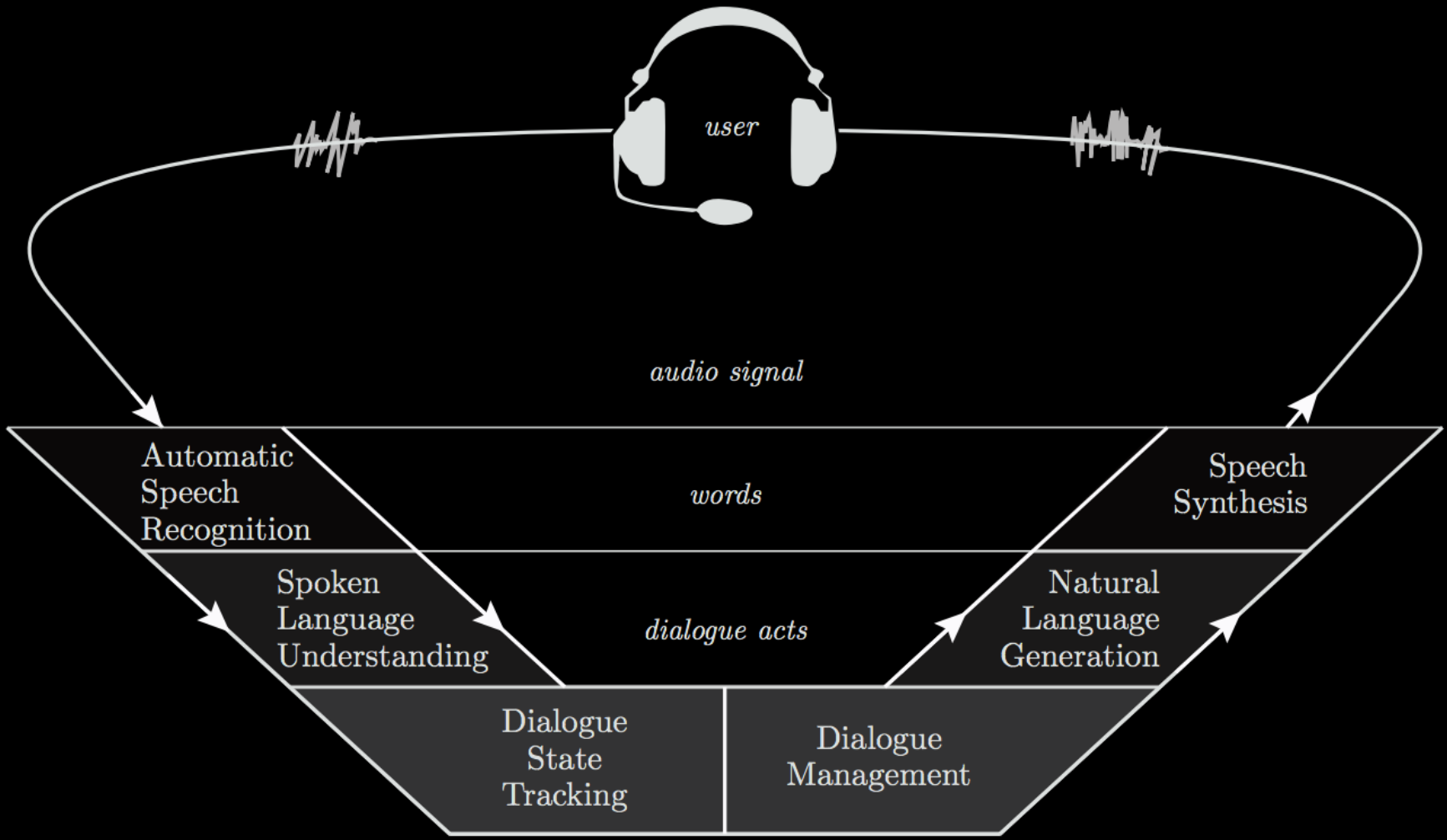
Task-Based Dialogue: (Useful) Virtual Assistants

Task-Based Dialogue Systems

Task-based dialogue systems help users achieve goals such as finding restaurants or booking flights.



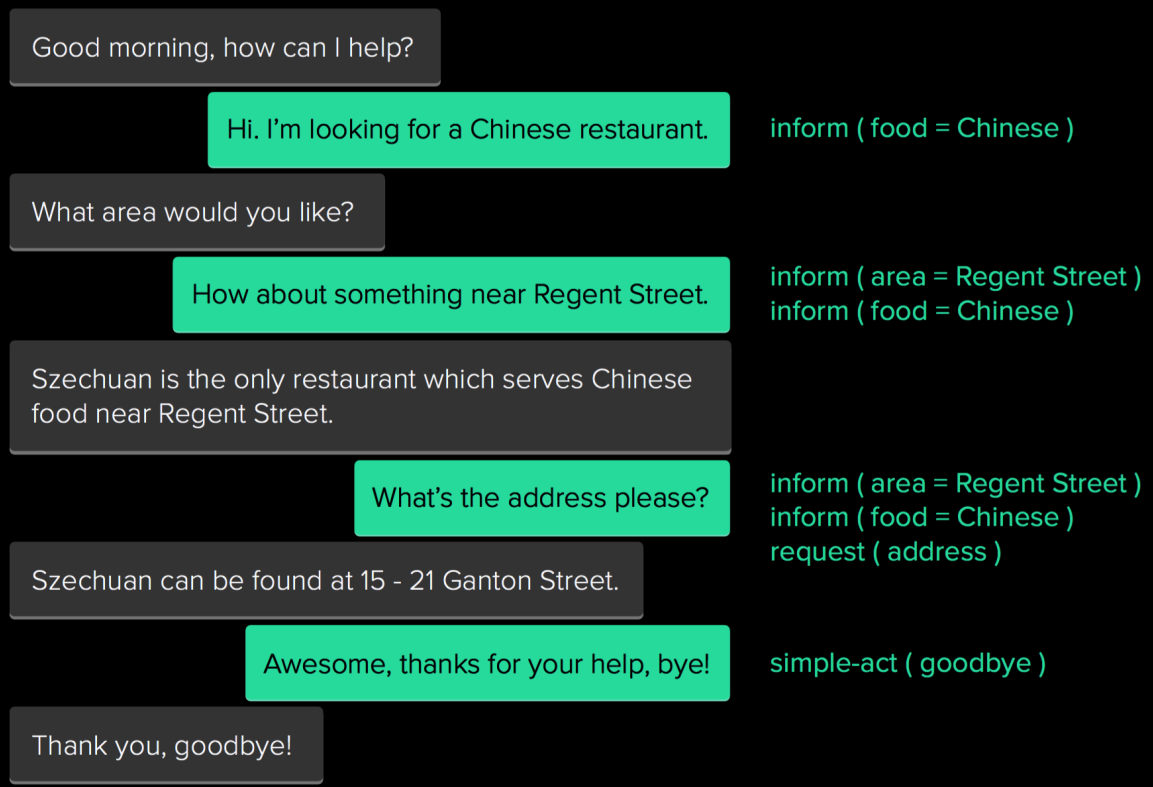
Modular Dialogue System Pipelines



Dialogue State Tracking

Task-Oriented Dialogue Systems are defined by their Domain Ontology

Task-oriented systems' ontologies consist of a collection of slots $s \in S$ (i.e. *food*, *price*, etc.) and their slot values V_s (*cheap*, *expensive*, etc.).



The Power of Delexicalised Features

Delexicalised Features

This model is powered by **delexicalised features:** all occurrences of slot names and/or slot values in an utterance are replaced with **generic tags**

The use of delexicalised n-gram features facilitates:

- Faster learning by facilitating transfer learning across slot values
- Generalisation to unseen slot values (or even entirely new slots)
- Bootstrapping dialogue systems to new domains with limited data

Shortcomings of Delexicalisation-Based Models

Delexicalised Features = Exact Matching

Given an arbitrary domain ontology, delexicalisation-based models provide data-efficient language understanding - as long as users use only the actual ontology values to express their search constraints!

User: I'm looking for an affordable restaurant
`inform(price=cheap)`
System: How about Thai food?
User: Yes please, in central Cambridge
`inform(price=cheap, food=Thai, area=centre)`
System: The House serves cheap Thai food
User: Where is it?
`inform(price=cheap, food=Thai, area=centre); request(address)`
System: The House is at 106 Regent Street

Traditional 'Solution': Semantic Dictionaries

Delexicalised Features = Exact Matching

Delexicalisation-based models allow fast deployment to new dialogue domains, but introduce a complete dependency on **semantic dictionaries**.

Food=Cheap: [affordable, budget, low-cost, low-priced, inexpensive, cheaper, economic, ...]

Rating=High: [best, high-rated, highly rated, top-rated, cool, chic, popular, trendy, ...]

Area=Centre: [center, downtown, central, city centre, midtown, town centre, ...]

A subsample of a semantic dictionary with rephrasings for three ontology values in a *restaurant search* domain akin to DSTC2.

Can we use word embeddings instead?

Distributional Hypothesis

Learning word embeddings from co-occurrence information in corpora coalesces the notions of *semantic similarity* and *conceptual association*.

Word	**east**	**expensive**	**British**
	west	pricey	**American**
	north	**cheaper**	**Australian**
	south	costly	Britain
	southeast	overpriced	**European (TBC)**
	northeast	**inexpensive**	England

Nearest neighbours using GloVe vectors

Semantic Specialisation using Linguistic Constraints

Semantically-Specialising Word Vector Spaces

Inject **antonym** and **synonym** constraints into pre-trained word vectors.

Word	**east**	**expensive**	**British**
Before	**west**	pricey	**American**
	north	**cheaper**	**Australian**
	south	costly	Britain
	southeast	overpriced	**European**
	northeast	**inexpensive**	England
After	eastward	costly	Brits
	eastern	pricy	London
	easterly	overpriced	BBC
	-	pricey	UK
	-	afford	Britain

Nearest neighbours before and after *counter-fitting* (Mrkšić et al., NAACL-16)

Semantic Specialisation: Motivation

Drawbacks of Popular Word Vector Collections

Unsupervised methods which induce vector representations from large textual corpora coalesce several types of information.

User: I'm looking for a cheaper restaurant
`inform(price=cheap)`
System: What kind of food?
User: English, in eastern Cambridge
`inform(price=cheap, food=British, area=east)`
System: The Green Man is the best choice
User: Where is it?
`inform(price=cheap, food=British, area=east); request(address)`
System: The Green Man is at 59 High St, Grantchester

Semantic Specialisation: Motivation

Drawbacks of Popular Word Vector Collections

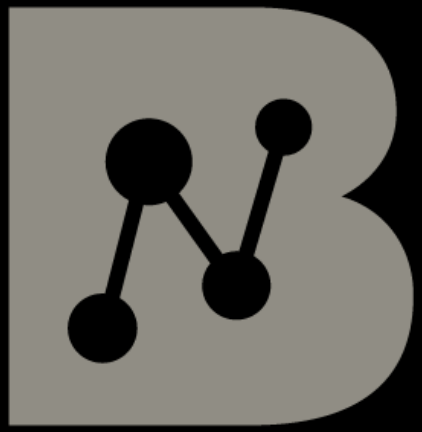
Unsupervised methods which induce vector representations from large textual corpora coalesce several types of information.

User: I'm looking for a cheaper restaurant
`inform(price=expensive)`
System: What kind of food?
User: English, in eastern Cambridge
`inform(price=expensive, food=Spanish, area=east)`
System: The Green Man is the best choice
User: Where is it?
`inform(price=expensive, food=Spanish, area=east);`
`request(address)`
System: The Green Man is at 59 High St, Grantchester

Moving Past Distributional Models

Making Use of Lexical Resources

Unsupervised methods making use of distributional information are both theoretically interesting and require no manual annotation. However, if our goal is optimising downstream performance, why not make use of all the lexical resources already available?



BabelNet

SEARCH, TRANSLATE, LEARN!

Paraphrase.org

Linguistic Constraints

In the **monolingual** scenario, we use linguistic constraints from a diverse collection of semantic lexicons:

Constraint	**Relation**	**Source**
(response, reply)	SYN	WordNet
(enemy, foe)	SYN	WordNet
(wait, anticipate)	SYN	BabelNet
(doctorate, postgraduate)	SYN	BabelNet
(costs, expense)	SYN	PPDB
(miserable, poor)	SYN	PPDB
(demand, supply)	ANT	WordNet
(stand, sit)	ANT	WordNet
(worthless, valuable)	ANT	BabelNet
(commencement, finishing)	ANT	BabelNet
(dishonour, honored)	ANT	PPDB
(intellect, stupidly)	ANT	PPDB

Retrofitting (Faruqui et al., NAACL 2015)

First Post-Processing Approach

Optimise a cost function which brings semantically similar words close together while keeping them (relatively) close to their initial distributional vectors.

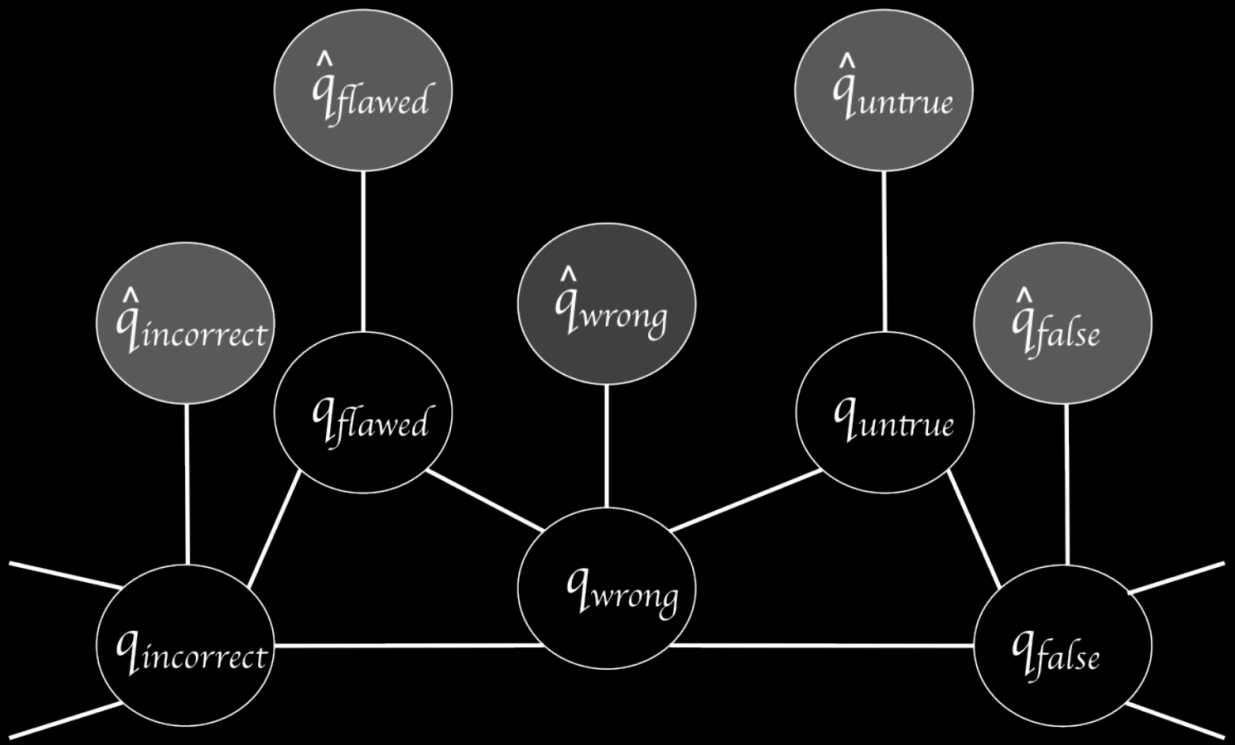
Let V be the vocabulary (with N words), and S the set of synonymous word pairs (e.g. *sophisticated* and *refined*). Let each word pair $(x_l, x_r) \in S$ correspond to vector pairs $(\mathbf{x}_l, \mathbf{x}_r)$. The retrofitting cost function is:

$$\Psi(V, S) = \sum_{x_i \in V} \left(||\mathbf{x}_i - \widehat{\mathbf{x}}_i||^2 + \sum_{(x_i, x_j) \in S} \beta_{i,j} ||\mathbf{x}_i - \mathbf{x}_j||^2 \right)$$

where $\beta_{i,j} = \frac{1}{deg(x_i)}$, and $deg(x_i)$ is the number of constraints in S which feature x_i. $\widehat{\mathbf{x}}_i$ is the initial distributional vector for x_i.

Retrofitting (Faruqui et al., NAACL 2015)

$$\Psi(V, S) = \sum_{x_i \in V} \left(||\mathbf{x}_i - \widehat{\mathbf{x}}_i||^2 + \sum_{(x_i, x_j) \in S} \beta_{i,j} ||\mathbf{x}_i - \mathbf{x}_j||^2 \right)$$



PARAGRAM (Wieting et al., 2015)

The PARAGRAM method (Wieting et al., 2015) improves on retrofitting by using a more sophisticated **"Attract"** term.

If S is again the set of synonymous word pairs, the procedure iterates over mini-batches of such constraints $\mathcal{B}_S$, optimising the following cost function:

$$S(\mathcal{B}_S) = \sum_{(x_l,x_r)\in\mathcal{B}_S} \left(ReLU\left(\delta_{sim} + \mathbf{x}_l\mathbf{t}_l - \mathbf{x}_l\mathbf{x}_r\right)\right. \\ \left. + \; ReLU\left(\delta_{sim} + \mathbf{x}_r\mathbf{t}_r - \mathbf{x}_l\mathbf{x}_r\right)\right)$$

where δ_{sim} is the similarity margin and $\mathbf{t}_l$ and $\mathbf{t}_r$ are **negative examples** for the given word pair (x_l, x_r).

PARAGRAM: the Attract Term

Negative Examples for each Synonymy Pair

For each synonymy pair $(\mathbf{x}_l, \mathbf{x}_r)$, the negative example pair $(\mathbf{t}_l, \mathbf{t}_r)$ is chosen from the remaining in-batch vectors so that $\mathbf{t}_l$ is the one closest (cosine similarity) to $\mathbf{x}_l$ and $\mathbf{t}_r$ is closest to $\mathbf{x}_r$.

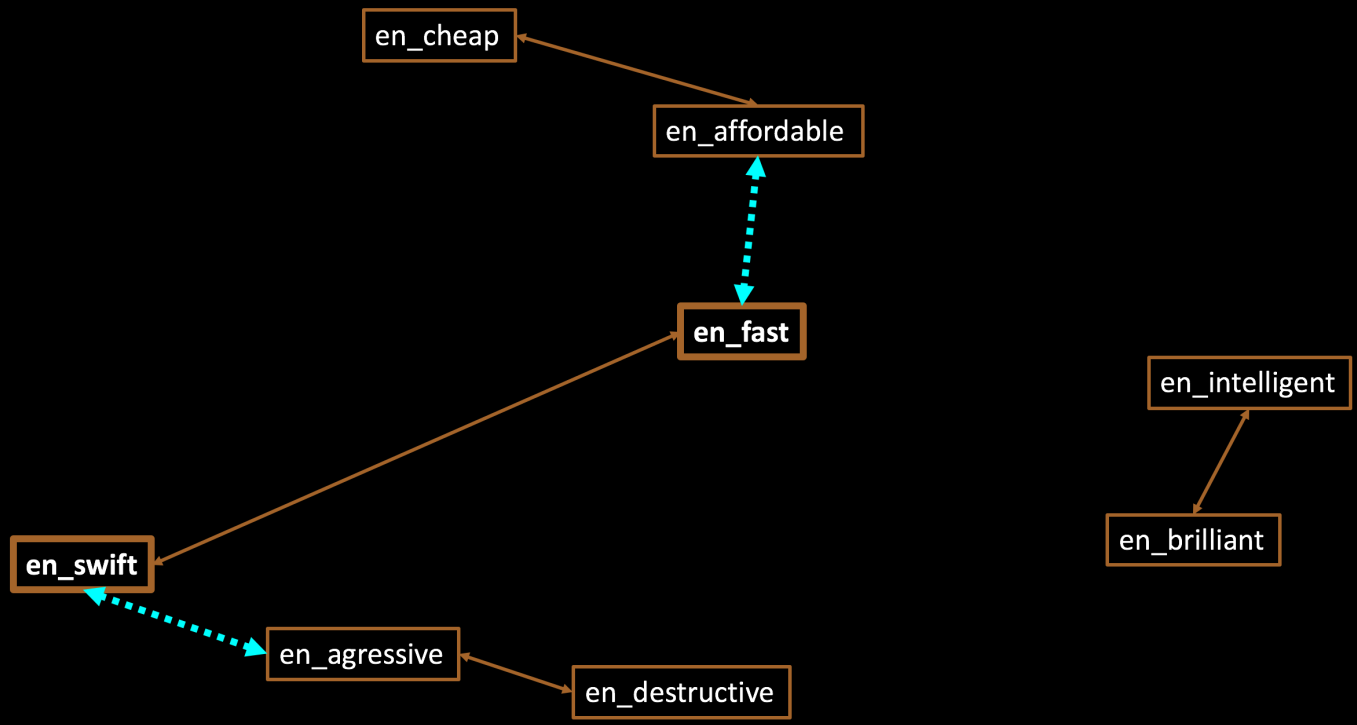
$$S(\mathcal{B}_S) = \sum_{(x_l, x_r) \in \mathcal{B}_S} \left(ReLU\left(\delta_{sim} + \mathbf{x}_l\mathbf{t}_l - \mathbf{x}_l\mathbf{x}_r\right)\right. \\ \left. + \; ReLU\left(\delta_{sim} + \mathbf{x}_r\mathbf{t}_r - \mathbf{x}_l\mathbf{x}_r\right)\right)$$

The two negative examples are used to force synonymous pairs to be closer to each other than to their respective negative examples (i.e. to any of the remaining words in the current mini-batch).

PARAGRAM: Negative Examples

Negative Examples for each Synonymy Pair

For each synonymy pair $(\mathbf{x}_l, \mathbf{x}_r)$, the negative example pair $(\mathbf{t}_l, \mathbf{t}_r)$ is chosen from the remaining in-batch vectors so that $\mathbf{t}_l$ is the one closest (cosine similarity) to $\mathbf{x}_l$ and $\mathbf{t}_r$ is closest to $\mathbf{x}_r$.



PARAGRAM: Regularisation

The second term tries to retain the beneficial semantic content embedded in the initial vector space.

L2 Regularisation

$$R(V) = \sum_{x_i \in V} \lambda_{reg} \left\| \widehat{\mathbf{x}}_i - \mathbf{x}_i \right\|_2$$

This term is near-identical to the one in retrofitting: here, λ_{reg} is fine-tuned to optimise performance. Again, it preserves semantic relations learned by distributional models that do not contradict the injected similarity constraints.

The Attract-Repel Model

The **Attract-Repel** model (Mrkšić et al., TACL 2017) extends the Paragram model with an additional **"Repel"** term.

The Repel Term

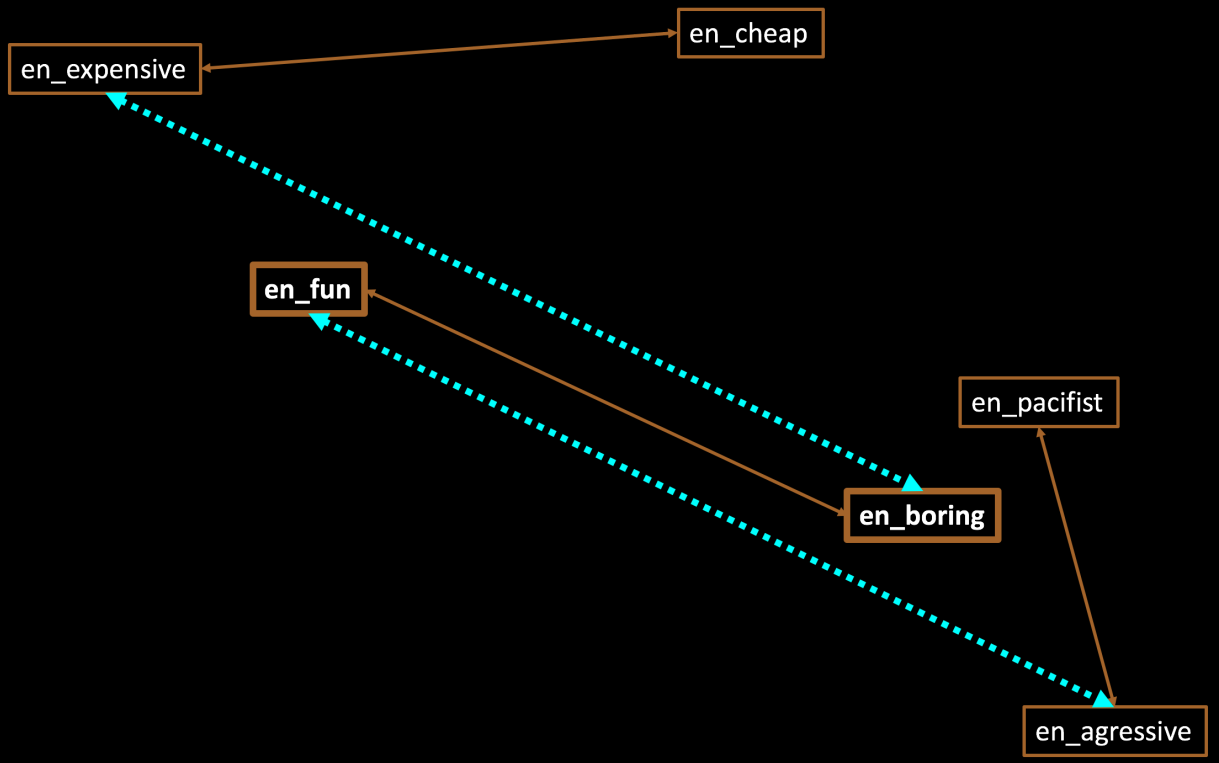
$$A(\mathcal{B}_A) = \sum_{(x_l,x_r)\in\mathcal{B}_A} \left(ReLU\left(\delta_{rpl} + \mathbf{x}_l\mathbf{x}_r - \mathbf{x}_l\mathbf{t}_r\right)\right.$$
$$\left. + \; ReLU\left(\delta_{rpl} + \mathbf{x}_l\mathbf{x}_r - \mathbf{x}_r\mathbf{t}_r\right)\right)$$

The **"repel"** term pushes words in undesirable relations (such as antonymy) away from each other in the reshaped vector space. These constraints can be monolingual (e.g., *en_brave* and *en_timid*) or cross-lingual (*en_peace* and *fr_guerre*).

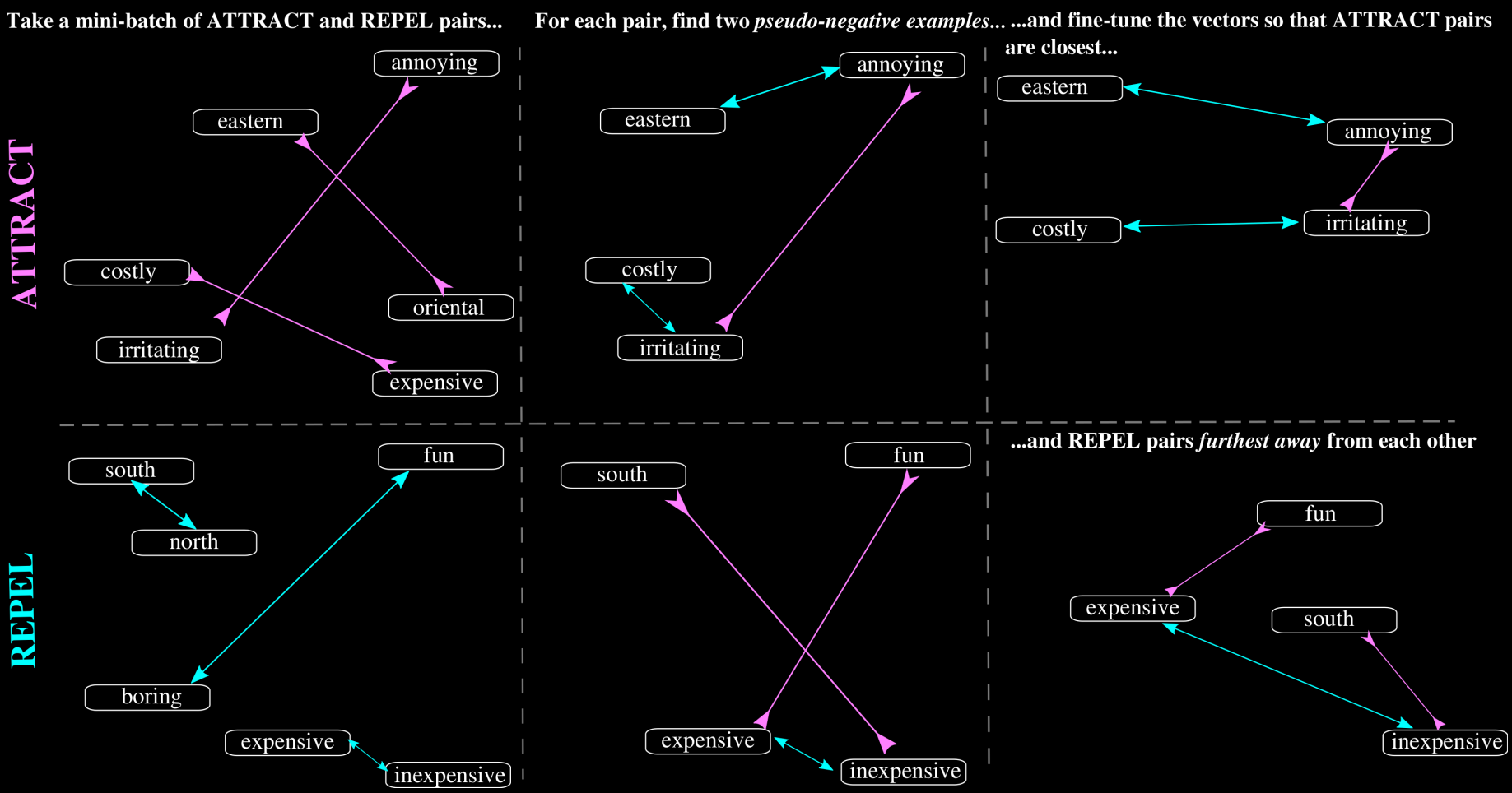
Repel Term: Negative Examples

Negative Examples for each Antonymy Pair

For each antonymy pair $(\mathbf{x}_l, \mathbf{x}_r)$, the negative example pair $(\mathbf{t}_l, \mathbf{t}_r)$ is chosen from the remaining in-batch vectors so that $\mathbf{t}_l$ is the one furthest away from $\mathbf{x}_l$ and $\mathbf{t}_r$ is the one furthest from $\mathbf{x}_r$.



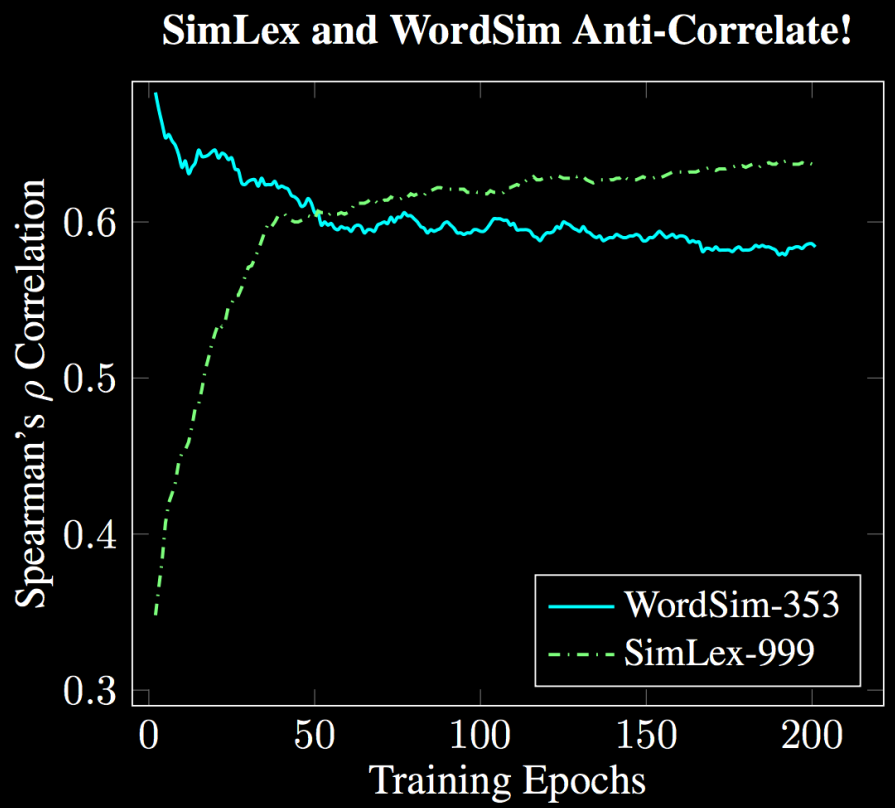
Attract-Repel in a Nutshell



Intrinsic Evaluation: WordSim-353 vs SimLex-999

Measuring Semantic Similarity (Properly)

WordSim-353 (and its similarity/relatedness splits) coalesce similarity and relatedness! Read (Hill et al., 2015) for a detailed explanation.



SimLex-999: State-of-the-Art

Model / Word Vectors	ρ
Neural MT Model (Hill et al., 2014)	0.52
Symmetric Patterns (Schwartz et al., 2015)	0.56
Non-distributional Vectors (Faruqui, 2015)	0.58
Skipgram-Retrofit (Kiela et al., 2015)	0.47
GloVe vectors (Pennington et al., 2014)	0.41
GloVe vectors + Retrofitting	0.53
GloVe + Counter-fitting	0.58
Paragram-WS353 (Wieting et al., 2015)	0.67
Paragram-WS353 + Retrofitting	0.67
Paragram-WS353 + Counter-fitting	0.72
dLCE (Nguyen et al., 2016)	0.59
charagram (Wieting et al., 2016)	0.71
Attract-Repel (Mrkšić et al., 2017)	0.75

SimLex-999 performance

Linguistic Constraints: Multilingual

In the **cross-lingual** scenario, we rely on BabelNet to extract cross-lingual constraints (synonymy and antonymy):

Constraint	Relation	Source
(en_sweet, it_dolce)	SYN	BabelNet
(en_work, fr_travail)	SYN	BabelNet
(en_school, de_schule)	SYN	BabelNet
(fr_montagne, de_gebirge)	SYN	BabelNet
(sh_gradonačelnik, en_mayor)	SYN	BabelNet
(nl_vrouw, it_donna)	SYN	BabelNet
(en_sour, it_dolce)	ANT	BabelNet
(en_asleep, fr_éveillé)	ANT	BabelNet
(en_cheap, de_teuer)	ANT	BabelNet
(de_langsam, es_despacio)	ANT	BabelNet
(sh_obeshrabriti, en_encourage)	ANT	BabelNet
(fr_jour, nl_nacht)	ANT	BabelNet

Cross-Lingual Semantic Specialisation

Cross-Lingual Vector Spaces

Constraint-based optimisation can be extended to cross-lingual specialisation. BabelNet constraints are used to bring the word vector spaces of various languages into a single unified vector space.

en_carpet			en_woman		
Slavic+EN	**Germanic**	**Romance+EN**	**Slavic + EN**	**Germanic**	**Romance+EN**
en_rug	de_teppichboden	en_rug	ru_женщина	de_frauen	fr_femme
bg_килим	nl_tapijten	it_moquette	bg_жените	sv_kvinnliga	en_womanish
ru_ковролин	en_rug	it_tappeti	sh_žena	sv_kvinna	es_mujer
bg_килими	de_teppich	pt_tapete	en_womanish	sv_kvinnor	pt_mulher
pl_dywany	en_carpeting	es_moqueta	bg_жена	de_weib	es_fémina
bg_мокет	de_teppiche	it_tappetino	pl_kobieta	en_womanish	en_womens
pl_dywanów	sv_mattor	en_carpeting	sh_treba	sv_kvinno	pt_feminina
sh_tepih	sv_matta	pt_carpete	bg_жени	de_frauenzimmer	pt_femininas
pl_wykładziny	en_carpets	pt_tapetes	en_womens	sv_honkön	es_femina
ru_ковер	nl_tapijt	fr_moquette	pl_kobiet	sv_kvinnan	fr_femelle
ru_коврик	nl_kleedje	en_carpets	sh_žene	nl_vrouw	pt_fêmea
sh_ćilim	nl_vloerbedekking	es_alfombra	pl_niewiasta	de_madam	fr_femmes
en_carpeting	de_brücke	es_alfombras	sh_žensko	sv_kvinnligt	it_donne
pl_dywan	de_matta	fr_tapis	sh_ženke	sv_gumman	es_mujeres
ru_ковров	nl_matta	pt_tapeçaria	pl_samica	sv_female	pt_fêmeas
en_carpets	en_mat	it_zerbino	ru_самка	sv_gumma	es_hembras

Attract-Repel: State-of-the-Art Performance

Word Vectors	English	German	Italian	Russian
Monolingual Distributional Vectors	0.32	0.28	0.36	0.38
Attract-Repel: Mono-Syn	0.56	0.40	0.46	0.53
Attract-Repel: Mono-Ant	0.42	0.30	0.45	0.41
Attract-Repel: Mono-Syn + Mono-Ant	0.65	0.43	0.56	0.56
Attract-Repel: Cross-Syn	0.57	0.53	0.58	0.46
Attract-Repel: Mono-Syn + Cross-Syn	0.61	0.58	0.59	0.54
Attract-Repel: All Constraints	**0.70**	**0.62**	**0.68**	**0.61**

Multilingual SimLex-999 performance of EN-DE-IT-RU vectors

All types of constraints are useful

Specialisation for Lower-Resource Languages?

Cross-Lingual Constraints - a Disambiguation Signal?

Even low resource-languages such as Irish Gaelic (GA) or Hebrew (HE) boost performance in resource-rich languages such as English!

	SimLex Languages				Non-SimLex, PPDB available						Non-SimLex, No PPDB					
	EN	**DE**	**IT**	**RU**	**NL**	**FR**	**ES**	**PT**	**PL**	**BG**	**SH**	**SV**	**HE**	**GA**	**VI**	**FA**
English	0.65	**0.69**	**0.70**	**0.70**	**0.70**	**0.72**	**0.72**	**0.70**	**0.70**	**0.68**	**0.69**	**0.70**	**0.66**	**0.67**	**0.67**	**0.68**
German	**0.61**	0.43	**0.58**	**0.56**	**0.55**	**0.60**	**0.59**	**0.56**	**0.54**	**0.52**	**0.53**	**0.55**	**0.50**	**0.49**	**0.48**	**0.51**
Italian	**0.69**	**0.65**	0.56	**0.64**	**0.67**	**0.68**	**0.68**	**0.66**	**0.66**	**0.62**	**0.63**	**0.63**	**0.59**	**0.60**	**0.58**	**0.61**
Russian	**0.63**	**0.59**	**0.62**	0.56	**0.61**	**0.61**	**0.62**	**0.58**	**0.60**	**0.61**	**0.59**	**0.60**	0.56	**0.57**	**0.58**	**0.58**

The effect on the four SimLex scores of cross-lingual semantic specialisation with each combination of the four SimLex languages with each of the sixteen languages. The four figures on the first diagonal indicate monolingual semantic specialisation for the SimLex languages. The figures in bold indicate improvements over these baselines.

Specialisation for Lower-Resource Languages?

Cross-Lingual Constraints - a Disambiguation Signal?

Even low resource-languages such as Irish Gaelic (GA) or Hebrew (HE) boost performance in resource-rich languages such as English!

	SimLex Languages				Non-SimLex, PPDB available						Non-SimLex, No PPDB					
	EN	**DE**	**IT**	**RU**	**NL**	**FR**	**ES**	**PT**	**PL**	**BG**	**SH**	**SV**	**HE**	**GA**	**VI**	**FA**
English	0.65	**0.69**	**0.70**	**0.70**	**0.70**	**0.72**	**0.72**	**0.70**	**0.70**	**0.68**	**0.69**	**0.70**	**0.66**	**0.67**	**0.67**	**0.68**
German	**0.61**	0.43	**0.58**	**0.56**	**0.55**	**0.60**	**0.59**	**0.56**	**0.54**	**0.52**	**0.53**	**0.55**	**0.50**	**0.49**	**0.48**	**0.51**
Italian	**0.69**	**0.65**	0.56	**0.64**	**0.67**	**0.68**	**0.68**	**0.66**	**0.66**	**0.62**	**0.63**	**0.63**	**0.59**	**0.60**	**0.58**	**0.61**
Russian	**0.63**	**0.59**	**0.62**	0.56	**0.61**	**0.61**	**0.62**	**0.58**	**0.60**	**0.61**	**0.59**	**0.60**	0.56	**0.57**	**0.58**	**0.58**

The effect on the four SimLex scores of cross-lingual semantic specialisation with each combination of the four SimLex languages with each of the sixteen languages. The four figures on the first diagonal indicate monolingual semantic specialisation for the SimLex languages. The figures in bold indicate improvements over these baselines.

Vectors for 51 EN-X bilingual vector spaces available at:

`github.com/nmrksic/attract-repel/`

Evaluation of Lower-Resource Languages?

Intrinsic Evaluation Datasets are Scarce

BabelNet provides an abundance of cross-Lingual semantic constraints. However, how do we know we are improving?

	Distrib.	+ EN	+ DE	+ IT	+ RU
Hebrew	0.28	**0.51**	0.46	0.52	0.45
Croatian	0.21	**0.62**	0.51	0.60	0.52
German	0.28	**0.58**	-	0.55	0.49
Italian	0.36	**0.69**	0.66	-	0.63
Russian	0.38	**0.56**	0.52	0.55	-

Bilingual semantic specialisation for five languages, with each row modelling the given language as low-resource and pairing it with three high-resource languages. Figures in bold indicate improvements over distributional vectors.

BabelNet 3.7: General Statistics

Number of languages:	271
Total number of Babel synsets:	13,801,844
Total number of Babel senses:	745,859,932
Total number of concepts:	6,066,396
Total number of Named Entities:	7,735,448
Total number of lexico-semantic relations:	380,239,084
Total number of glosses (textual definitions):	40,709,194
Total number of images:	10,767,833
Total number of Babel synsets with at least one domain:	2,675,385
Total number of compounds:	743,296
Total number of other forms:	6,393,568
Total number of Babel synsets with at least one picture:	2,948,668
Total number of RDF triples:	1,971,744,856

Back to Dialogue State Tracking

Neural Belief Tracker

How can we use the semantically specialised vector spaces in downstream applications such as Dialogue State Tracking?

NBT: Data-Driven End-to-End DST

Semantic Dictionaries Re-Introduce the SLU module

Semantic dictionaries can be hand-crafted or learned - but only for simple toy domains such as DSTC2. Moreover, the Amazon Mechanical Turk data collection framework forces the users to use very simple (and unnatural) language, understating the challenge of dealing with linguistic variation.

NBT: Data-Driven End-to-End DST

Semantic Dictionaries Re-Introduce the SLU module

Semantic dictionaries can be hand-crafted or learned - but only for simple toy domains such as DSTC2. Moreover, the Amazon Mechanical Turk data collection framework forces the users to use very simple (and unnatural) language, understating the challenge of dealing with linguistic variation.

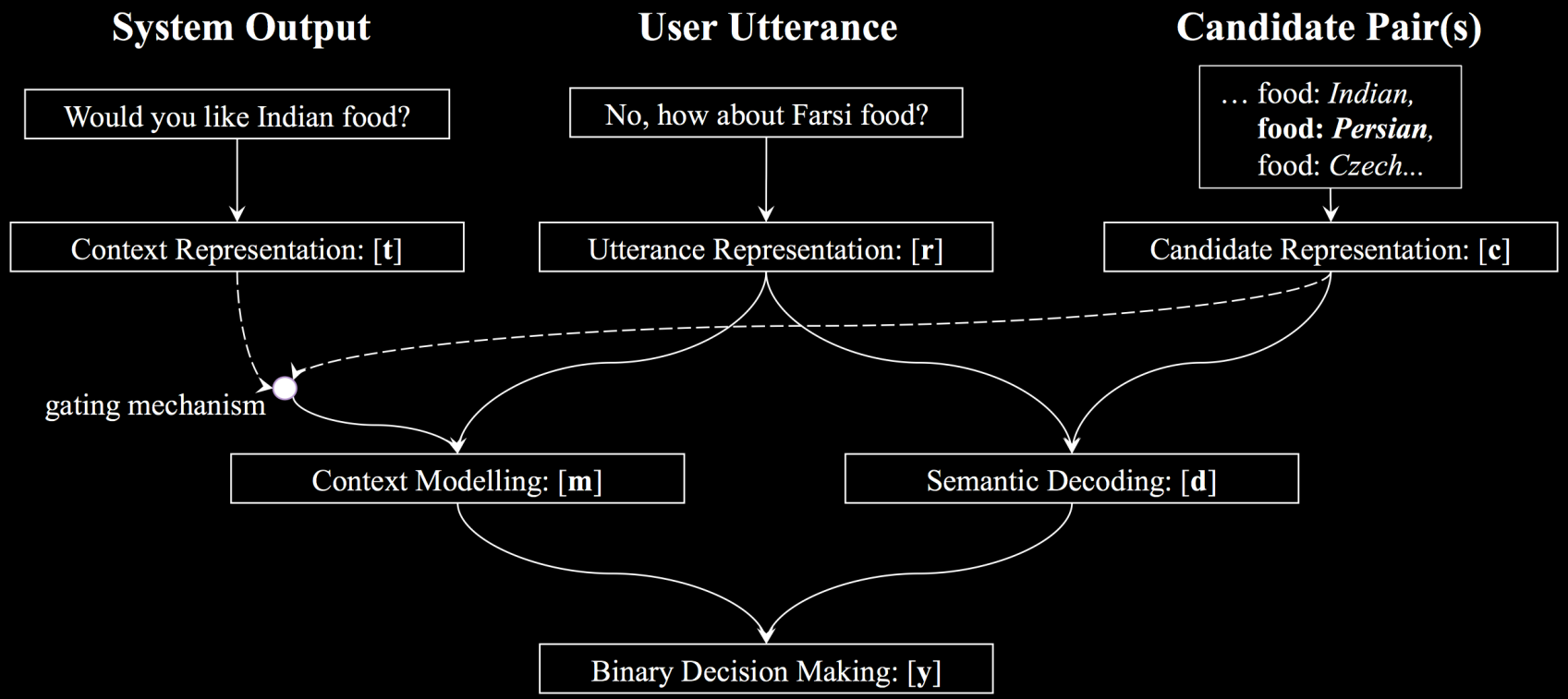
The **Neural Belief Tracker** (Mrkšić et al., ACL 2017) is a novel DST model/framework which aims to satisfy the following design goals:

- End-to-end learnable (no SLU modules or semantic dictionaries).
- Generalisation to unseen slot values.
- Capability of leveraging the semantic content of pre-trained word vector spaces without human supervision.

The Neural Belief Tracking (NBT) Framework

Representation Learning + Label Embedding + Binary Classification

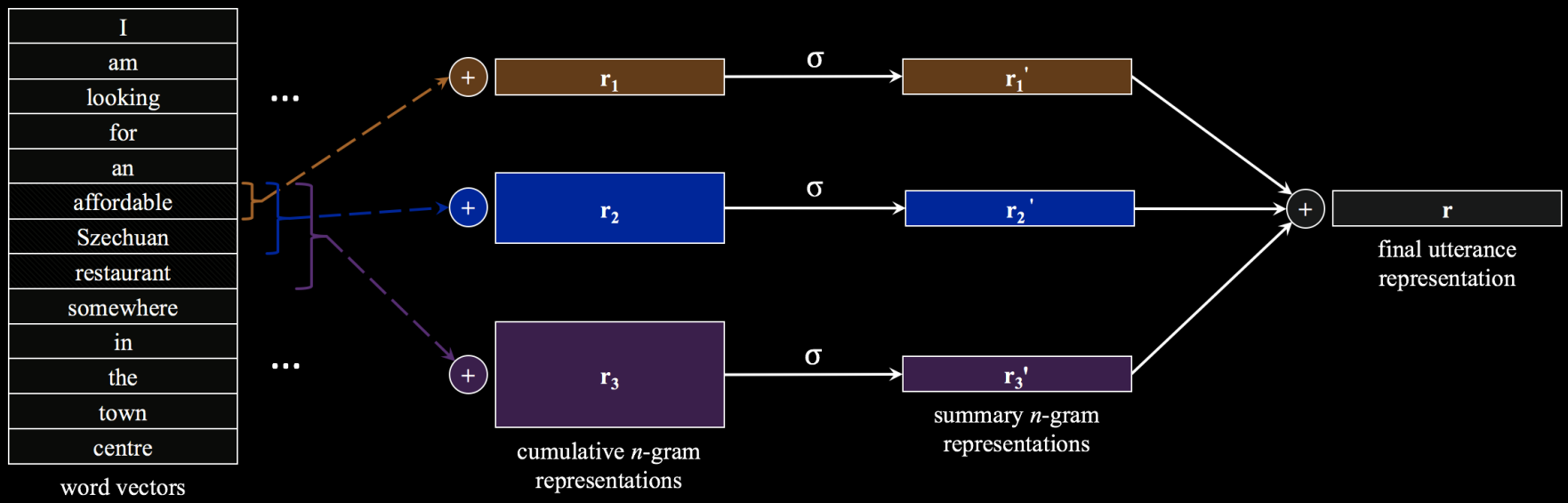
To overcome data sparsity, NBT models use *label embedding* to decompose the multi-class classification problem into many binary ones.



Representation Learning: the NBT-DNN Model

Let u represent a user utterance consisting of k_u words $u_1, u_2, \ldots, u_{k_u}$.
Each word has an associated **fixed** word vector $\mathbf{u}_1, \ldots, \mathbf{u}_{k_u}$.

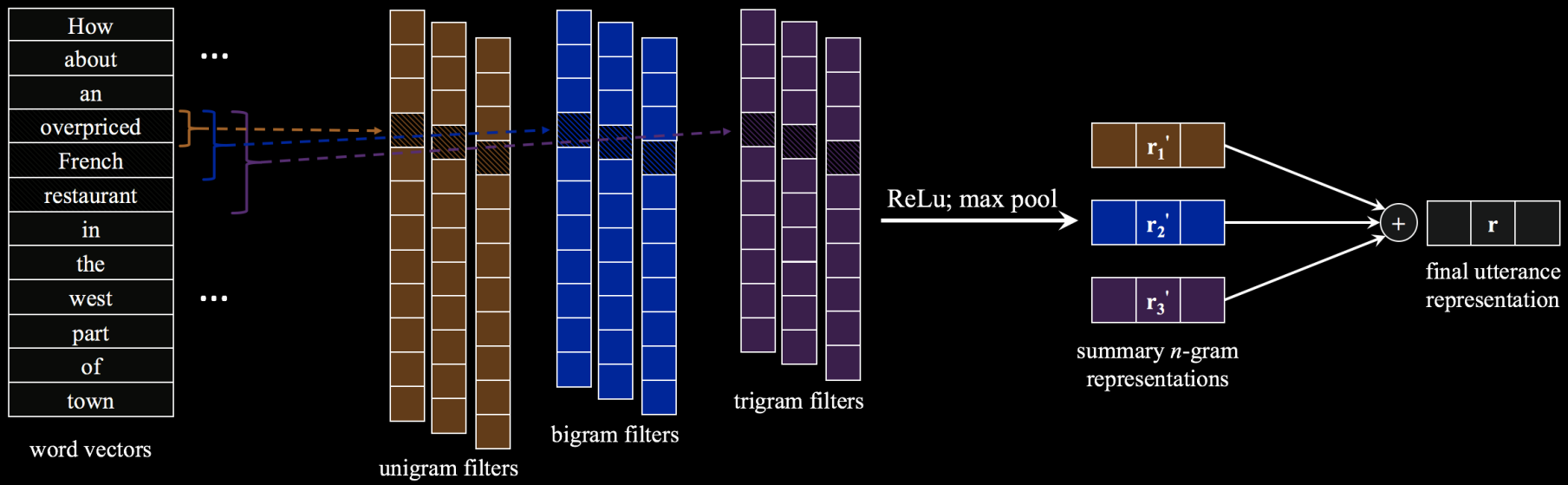
$$\mathbf{v}_i^n = \mathbf{u}_i \oplus \ldots \oplus \mathbf{u}_{i+n-1}$$



$$\mathbf{r}_n = \sum_{i=1}^{k_u-n+1} \mathbf{v}_i^n; \qquad \mathbf{r}'_n = \sigma(W_n^s \mathbf{r}_n + b_n^s); \qquad \mathbf{r} = \mathbf{r}'_1 + \mathbf{r}'_2 + \mathbf{r}'_3$$

Representation Learning: the NBT-CNN Model

Let $F_n^s \in R^{L \times nD}$ denote the collection of filters for each value of n, where $D = 300$ is the word vector dimensionality. If $\mathbf{v}_i^n$ denotes the concatenation of n **fixed** word vectors starting at index i, let $\mathbf{m}_n = [\mathbf{v}_1^n; \mathbf{v}_2^n; \ldots; \mathbf{v}_{k_u - n + 1}^n]$ be the list of n-grams that convolutional filters of length n run over.

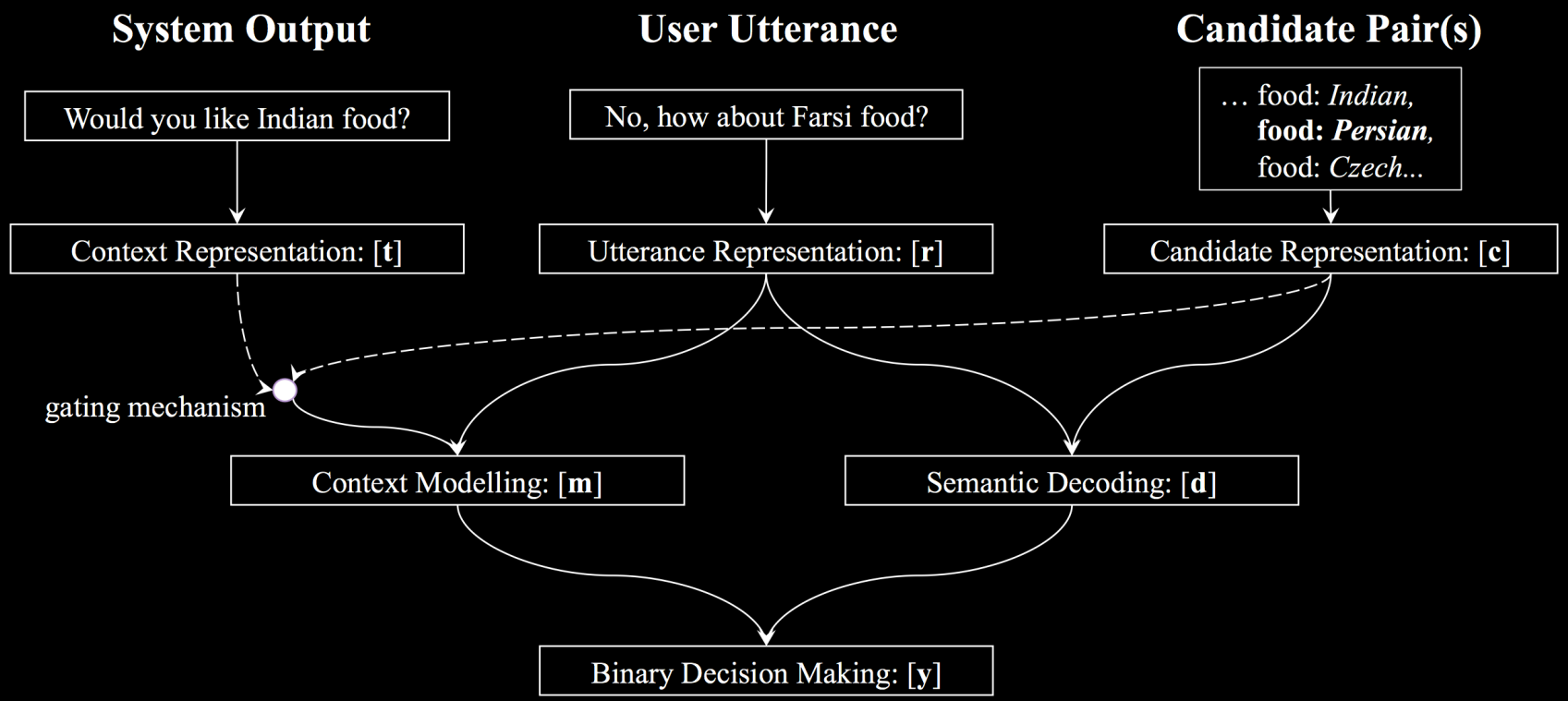


$$R_n = F_n^s \ \mathbf{m}_n \qquad \mathbf{r}'_n = \texttt{maxpool}\left(\texttt{ReLu}\left(R_n + b_n^s\right)\right) \qquad \mathbf{r} = \mathbf{r}'_1 + \mathbf{r}'_2 + \mathbf{r}'_3$$

How do we use the utterance representation?

Multi-class classification problem as many binary ones

We iterate over all slot-value pairs for the given slot, deciding whether each of them has been expressed in the given utterance.



Semantic Decoding

Let the vector space representations of a candidate pair's slot name and value be given by $\mathbf{c_s}$ and $\mathbf{c_v}$. The NBT framework learns to map these into a *candidate pair* representation $\mathbf{c}$ (of the same dimensionality as $\mathbf{r}$):

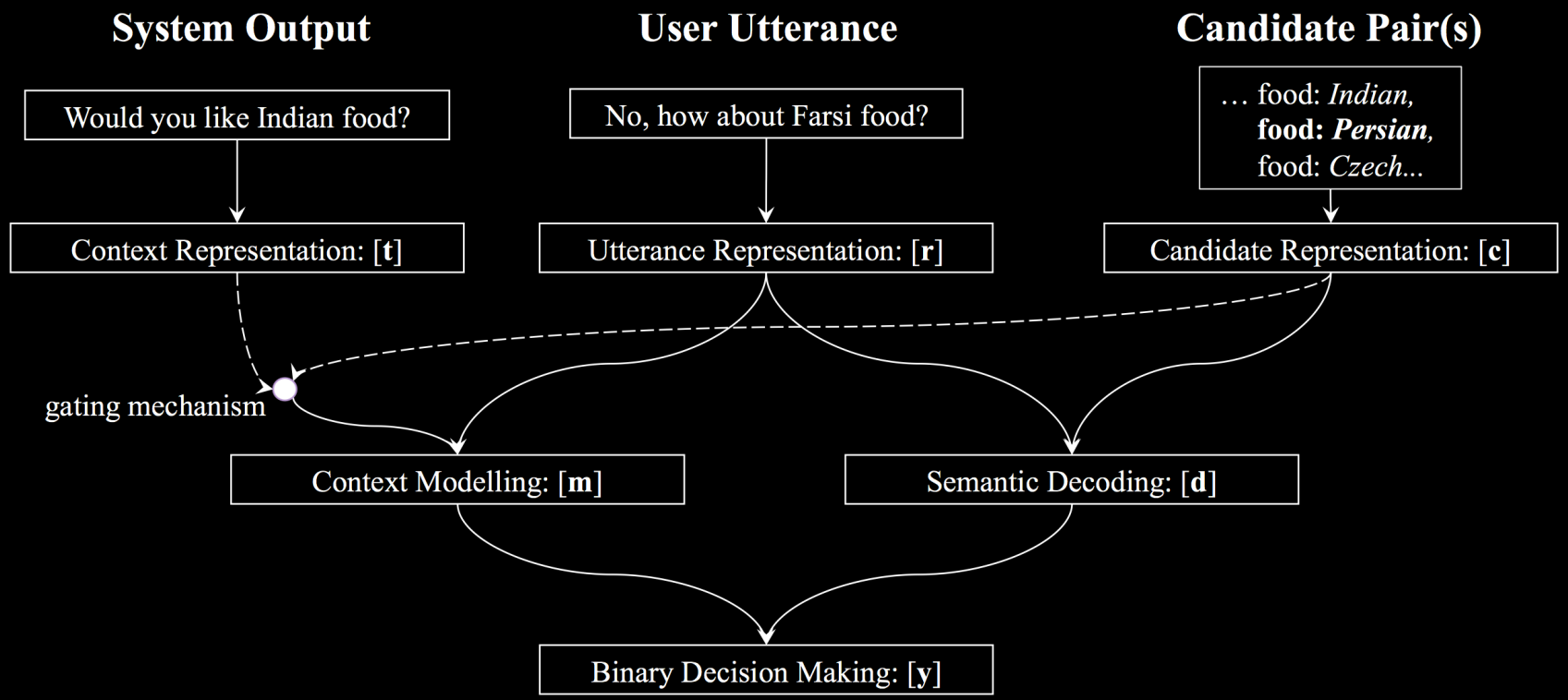
$$\mathbf{c} = \sigma\left(W_c^s(\mathbf{c_s} + \mathbf{c_v}) + b_c^s\right)$$
$$\mathbf{d} = \mathbf{r} \otimes \mathbf{c}$$

where $\otimes$ denotes *element-wise* vector multiplication. This vector is then passed to the downstream network, which combines it with the surrounding dialogue context to make a decision regarding the current candidate pair.

How do we take the system acts into account?

Gating Mechanisms for Modelling Context

Gating mechanisms activate the part of the network architecture which models context (limited to previous system acts).



Context Modelling

All previous system/user utterances are important, but the most relevant one is the last system utterance, in which the dialogue system could have performed (among others) one of the following two *system acts*:

- System Request: *'What price range would you like?'*
- System Confirm: *'How about Turkish food?'*

Context Modelling

Let $\mathbf{t_q}$ and $(\mathbf{t_s}, \mathbf{t_v})$ be the word vector representations of the arguments for the system request and confirm acts (zero vectors if none). The model computes the following measures of similarity between the system acts, candidate pair $(\mathbf{c_s}, \mathbf{c_v})$ and utterance representation $\mathbf{r}$:

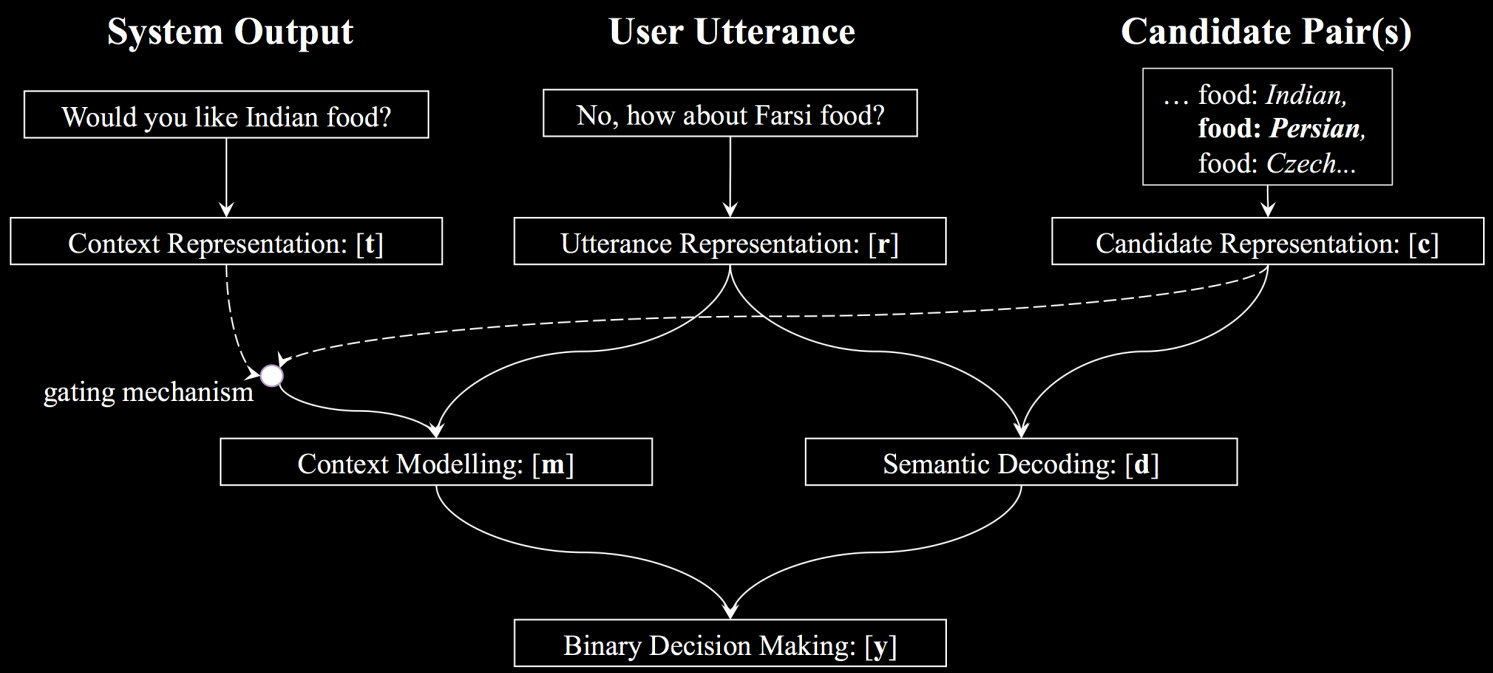
$$\begin{aligned}\mathbf{d_r} &= (\mathbf{c_s} \cdot \mathbf{t_q})\mathbf{r} \\ \mathbf{d_c} &= (\mathbf{c_s} \cdot \mathbf{t_s})(\mathbf{c_v} \cdot \mathbf{t_v})\mathbf{r} \\ \mathbf{m} &= \mathbf{d_r} \oplus \mathbf{d_c}\end{aligned}$$

where $\cdot$ denotes dot product and $\oplus$ denotes vector concatenation. The computed similarity terms act as gating mechanisms which only pass the utterance representation through if the system asked about the current candidate slot or slot-value pair.

What is the (final) NBT model output?

The NBT makes binary predictions for each slot value

Given a slot s with a set of values $v \in V_s$, the NBT models estimate $\mathbb{P}(s, v \mid u, sys)$, i.e. the probability that the utterance u which follows the system acts sys expresses each of the slot values.



Belief State Updates

Turn-level belief state estimate is then combined with the (cumulative) belief state up to time $(t-1)$ to get the updated belief state estimate:

$$\begin{aligned}\mathbb{P}(s, v \mid h^{1:t}, sys^{1:t-1}) \;=\; & \lambda\, \mathbb{P}\left(s, v \mid h^{t}, sys^{t-1}\right) \\ & + (1-\lambda)\, \mathbb{P}\left(s, v \mid h^{1:t-1}, sys^{1:t-2}\right)\end{aligned}$$

where λ is the coefficient which determines the relative weight of the turn-level and previous turns' belief state estimates. $\lambda = 0.55$ achieved the best performance on the DSTC 2 development set.

Dialogue State Tracking: Evaluation

We focus on two key evaluation metrics (Henderson et al., 2014):

- Goals (joint goal accuracy): the proportion of dialogue turns where all the user's search goal constraints were correctly identified;
- Requests: similarly, the proportion of dialogue turns where user's requests for information were identified correctly.

NBT Evaluation: DSTC2 and WOZ 2.0

WOZ 2.0

1,200 dialogues collected using the Wizard-of-Oz setup; users typed instead of using speech, giving them freedom to use more sophisticated language.

(Wen et al., EACL 2017; Mrkšić et al., TACL 2017)

NBT Evaluation: DSTC2 and WOZ 2.0

WOZ 2.0

1,200 dialogues collected using the Wizard-of-Oz setup; users typed instead of using speech, giving them freedom to use more sophisticated language.

(Wen et al., EACL 2017; Mrkšić et al., TACL 2017)

Model	DSTC2		WOZ 2.0	
	Goals	Requests	Goals	Requests
Baseline DST	69.1	95.7	70.8	87.1
+ sem. dict.	72.9*	95.7	83.7*	87.6
NBT-DNN	72.6*	96.4	**84.4***	91.2*
NBT-CNN	**73.4***	**96.5**	84.2*	**91.6***

DSTC2 and WOZ 2.0 test set accuracies for: **a)** joint goals; and **b)** turn-level requests. The asterisk indicates statistically significant improvement over the baseline delexicalisation-based trackers (paired t-test; $p < 0.05$).

The Importance of Semantic Specialisation

Three different word vector collections: **1)** 'random' word vectors initialised using the xavier initialisation; **2)** distributional GloVe vectors; and **3)** *semantically specialised* Paragram-SL999 vectors.

Word Vectors	DSTC2		WOZ 2.0	
	Goals	Requests	Goals	Requests
xavier	64.2	81.2	81.2	90.7
GloVe	69.0*	96.4*	80.1	91.4
Paragram-SL999	**73.4***	**96.5***	**84.2***	**91.6**

Specialisation is crucial

Dialogue State Tracking in Italian and German

Multilingual WOZ 2.0; Italian and German

The 1,200 dialogues in WOZ 2.0 were translated by native Italian and German speakers instructed to consider preceding dialogue context.

(Mrkšić et al., TACL 2017)

Dialogue State Tracking in Italian and German

Multilingual WOZ 2.0; Italian and German

The 1,200 dialogues in WOZ 2.0 were translated by native Italian and German speakers instructed to consider preceding dialogue context.

(Mrkšić et al., TACL 2017)

Word Vectors	EN	DE	IT
Monolingual Distributional Vectors	0.32	0.28	0.36
Attract-Repel: Mono-Lingual Spec.	0.65	0.43	0.56
Attract-Repel: Mono.+Cross-Ling. Spec.	**0.70**	**0.62**	**0.68**

Multilingual SimLex-999 performance of EN-DE-IT-RU vectors.

Dialogue State Tracking in Italian and German

Multilingual WOZ 2.0; Italian and German

The 1,200 dialogues in WOZ 2.0 were translated by native Italian and German speakers instructed to consider preceding dialogue context.

(Mrkšić et al., TACL 2017)

Word Vectors	EN	DE	IT
Monolingual Distributional Vectors	0.32	0.28	0.36
Attract-Repel: Mono-Lingual Spec.	0.65	0.43	0.56
Attract-Repel: Mono.+Cross-Ling. Spec.	**0.70**	**0.62**	**0.68**

Multilingual SimLex-999 performance of EN-DE-IT-RU vectors.

Word Vector Space	EN	IT	DE
Monolingual Distributional Vectors	77.6	71.2	46.6
+ Monolingual Specialisation	80.9	72.7	52.4
++ Cross-Lingual Specialisation	80.3	75.3	55.7

Bootstrapping DST for Resource-Poor Languages

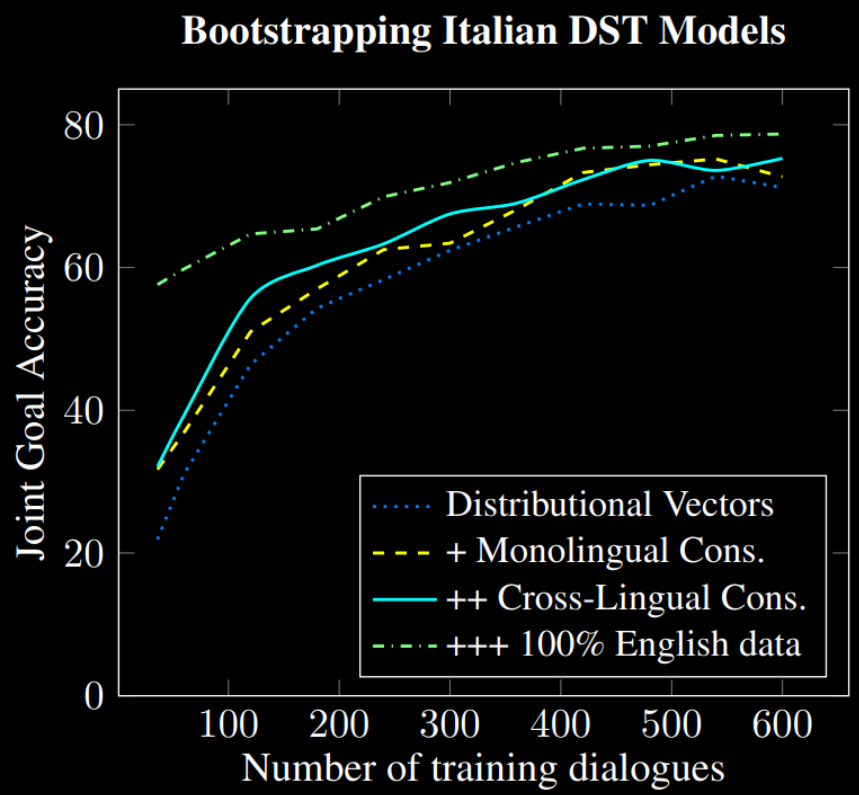
Ontology Grounding: Multilingual DST Models

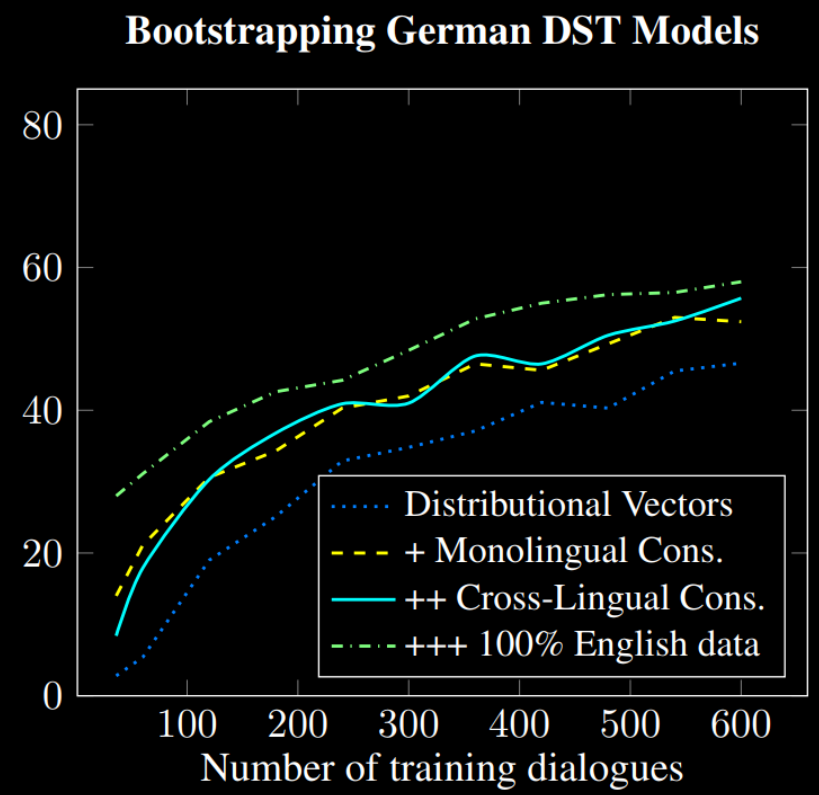
The dialogue domain ontology (i.e. the concepts it expresses) is language agnostic, which means that 'labels' persist across languages. Given training data for two (or more) languages, and a cross-lingual vector space of high quality, we train the first-ever *multilingual* DST model.

Bootstrapping DST for Resource-Poor Languages

Ontology Grounding: Multilingual DST Models

The dialogue domain ontology (i.e. the concepts it expresses) is language agnostic, which means that 'labels' persist across languages. Given training data for two (or more) languages, and a cross-lingual vector space of high quality, we train the first-ever *multilingual* DST model.

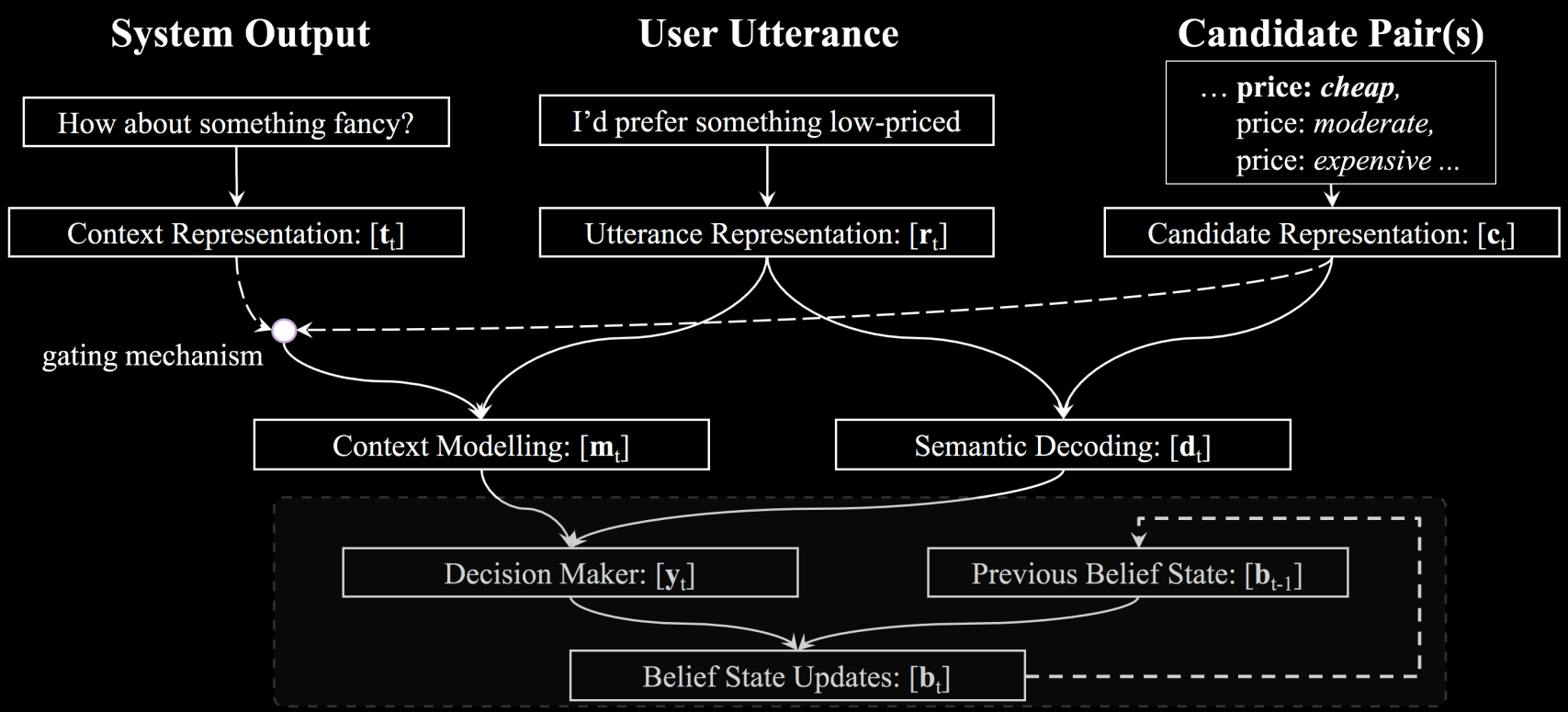




NBTv2: Fully Statistical Neural Belief Tracker

Learned Belief State Updates...

...instead of resorting to hand-crafted programmatic rules



(Mrkšić and Vulić, ACL 2018)

NBTv2: Learned Belief State Update Mechanisms

$$\mathbf{b}_s^t = \phi(\mathbf{y}_s^t, \mathbf{b}_s^{t-1})$$

1. One-Step Markovian Update

$$\mathbf{b}_s^t = softmax\left(W_{curr}\mathbf{y}_s^t + W_{past}\mathbf{b}_s^{t-1}\right) \quad (1)$$

This variant violates the NBT design paradigm: each row of the two matrices learns to operate over *specific* slot values. This means the model will not learn to predict or maintain slot values as part of the belief state if it has not encountered these values during training.

NBTv2: Learned Belief State Update Mechanisms

$$\mathbf{b}_s^t = \phi(\mathbf{y}_s^t, \mathbf{b}_s^{t-1})$$

2. Constrained Markovian Update

$$W_{curr,i,j} = \begin{cases} a_{curr}, & \text{if } i = j \\ b_{curr}, & \text{otherwise} \end{cases}$$

$$W_{past,i,j} = \begin{cases} a_{past}, & \text{if } i = j \\ b_{past}, & \text{otherwise} \end{cases}$$

This variant constrains the two matrices so that each of them contains only two different scalar values. The parameters acting over all slot values are in this way tied, ensuring that the model can deal with slot values unseen in training.

NBTv2: Learned Belief State Update Mechanisms

It is easier to deploy NBTv2 to different domains and other languages: no hand-crafting involved!

	English WOZ 2.0	
Model Variant	glove (dist)	paragram-sl999
Rule-Based	80.1	84.2
1. One-Step	80.8	82.1
2. Constrained	**81.8**	**84.8**

	Italian WOZ 2.0		German WOZ 2.0	
	dist	**spec**	**dist**	**spec**
Rule-Based Update	**74.2**	76.0	60.6	66.3
Learned Update	73.7	**76.1**	**61.5**	**68.1**

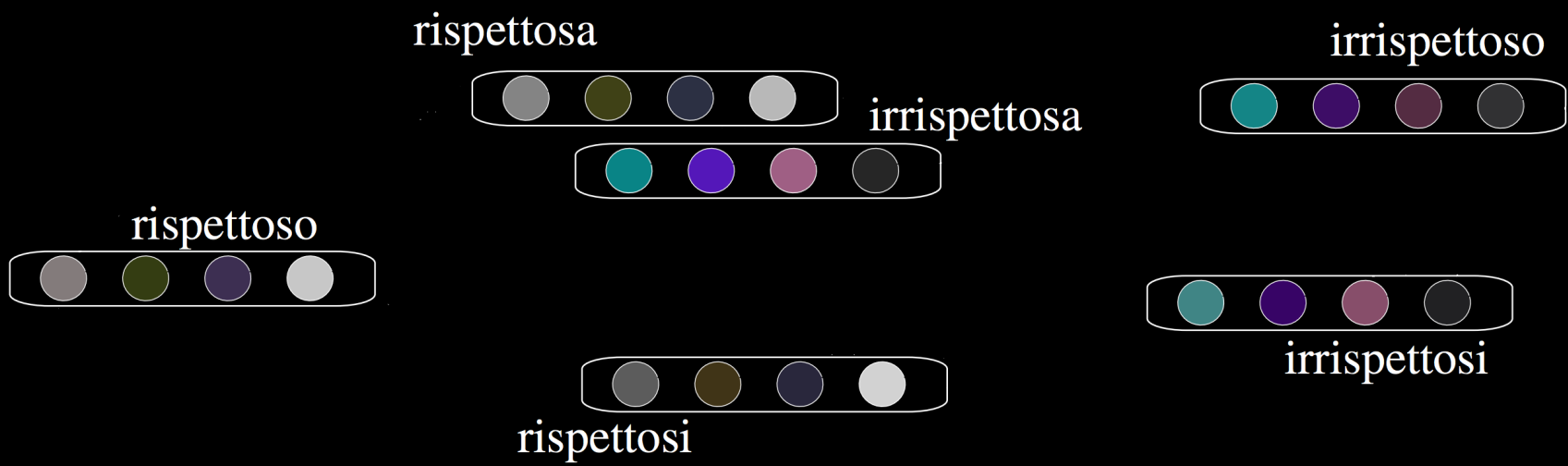
Word Vectors and Morphology

1. Estimating Rare Words

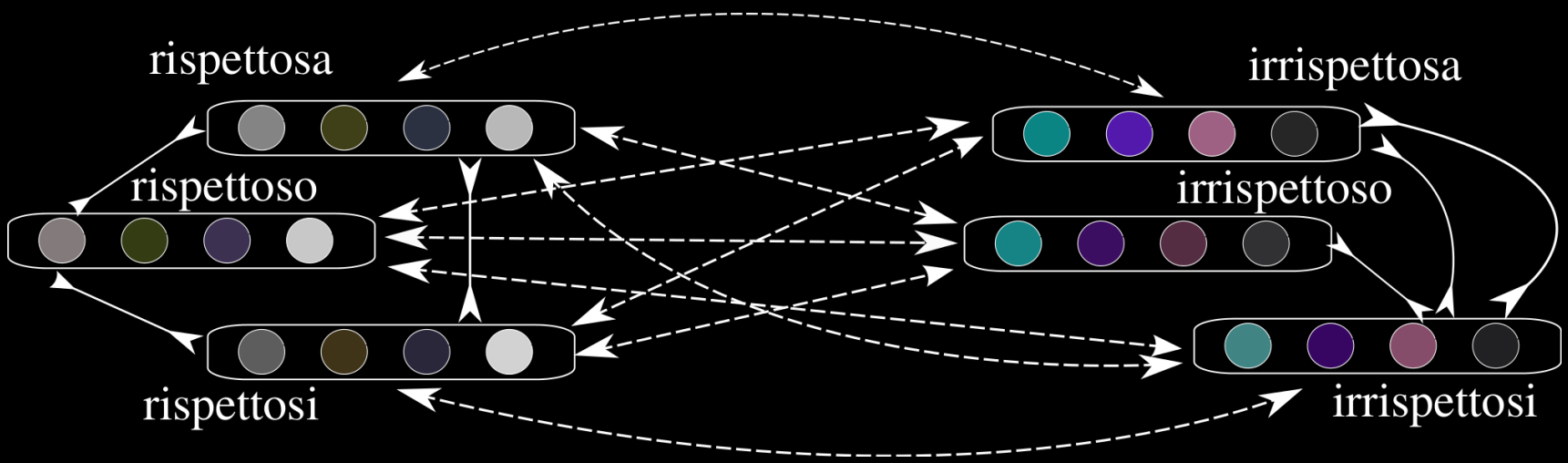
A single lemma can have many different surface realisations

2. Making Use of Sub-Word Semantics

Morphological phenomena provide an inexpensive source of supervision. Consider both *inflectional* and *derivational* forms.



Morph-fitting: Illustration



(Vulić et al., ACL 2017)

Morphological Forms - Nearest Neighbours?

	en_slow	**de_langsam**	**it_lento**
Before	fast	allmählich	lentissimo
	slower	rasch	lenta
	slower	gemächlich	inesorabile
	slowed	schnell	rapidissimo
	slowing	explosionsartig	graduale
After	slow	langsamer	lenti
	slowing	langsames	lente
	slowed	langsame	lenta
	slowness	langsamem	veloce
	slows	langsamen	rapido

Morph-fitting: Illustration

Word	Inflectional Synonyms	Derivational Antonyms
mature		

Morph-fitting: Illustration

Word	Inflectional Synonyms	Derivational Antonyms
mature	matureed matureing matures matured maturing	

Morph-fitting: Illustration

Word	Inflectional Synonyms	Derivational Antonyms
mature	~~matureed~~ ~~matureing~~ **matures** **matured** **maturing**	

Synonyms: (mature, matures), (mature, matured), (mature, maturing)

Morph-fitting: Illustration

Word	Inflectional Synonyms	Derivational Antonyms
	~~matureed~~	dismature
	~~matureing~~	ilmature
	matures	unmature
	matured	inmature
mature	**maturing**	immature
		irmature
		mismature
		nonmature
		antimature

Synonyms: (mature, matures), (mature, matured), (mature, maturing)

Morph-fitting: Illustration

Word	Inflectional Synonyms	Derivational Antonyms
	~~matureed~~	~~dismature~~
	~~matureing~~	~~ilmature~~
	matures	~~unmature~~
	matured	~~inmature~~
mature	**maturing**	**immature**
		~~irmature~~
		~~mismature~~
		~~nonmature~~
		~~antimature~~

Synonyms: (mature, matures), (mature, matured), (mature, maturing)

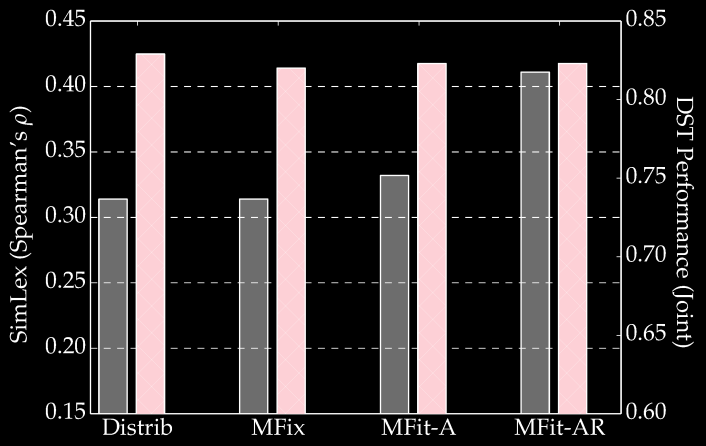
Antonyms: (mature, immature)

Semantic Specialisation with Morphology

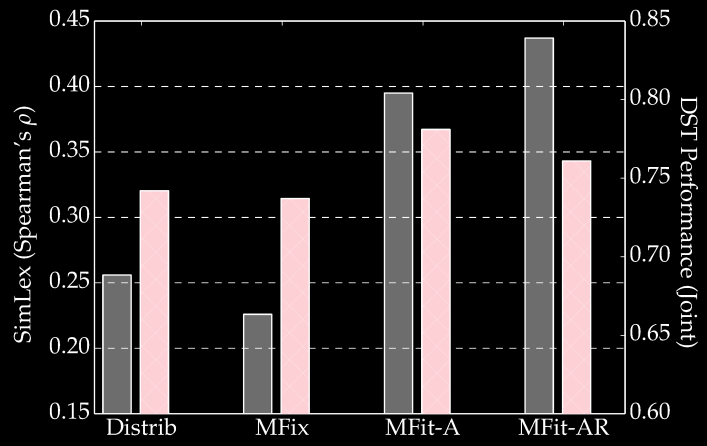
Simple morphological rules yield thousands of constraints:

Language	Constraint	Relation	Rule
EN	(sunflower, sunflowers)	SYN	Singular-Plural
	(suffer, suffered)	SYN	Past Participle
	(ambiguous, unambiguous)	ANT	Derivational Antonymy
	(regular, irregular)	ANT	Derivational Antonymy
IT	(zucchero, zuccheri)	SYN	Singular-Plural
	(vincere, vincono)	SYN	Conjugation
	(rapido, rapida)	SYN	Gender
	(visibilità, invisibilità)	ANT	Derivational Antonymy
DE	(Kategorie, Kategorien)	SYN	Singular-Plural
	(kaufst, kauft)	SYN	Conjugation
	(katalanisch, katalanischem)	SYN	Declension
	(dokumentiert, undokumentiert)	ANT	Derivational Antonymy

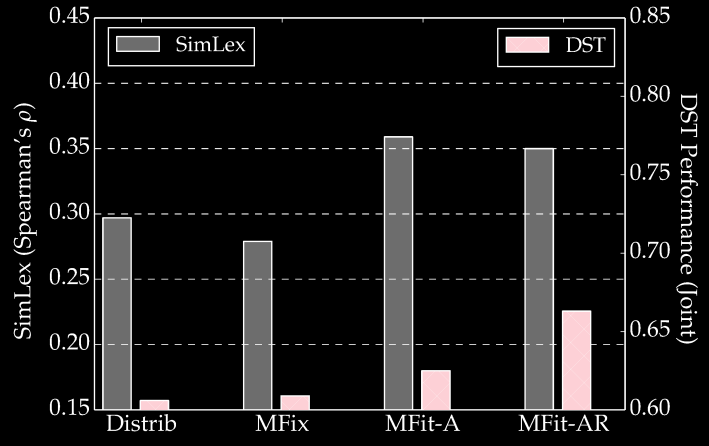
Morph-Fitting to the Rescue



(a) **English**



(b) **Italian**



(c) **German**

(Full Vocabulary) Post-Specialisation

Post-processors such as Attract-Repel fine-tune only seen words

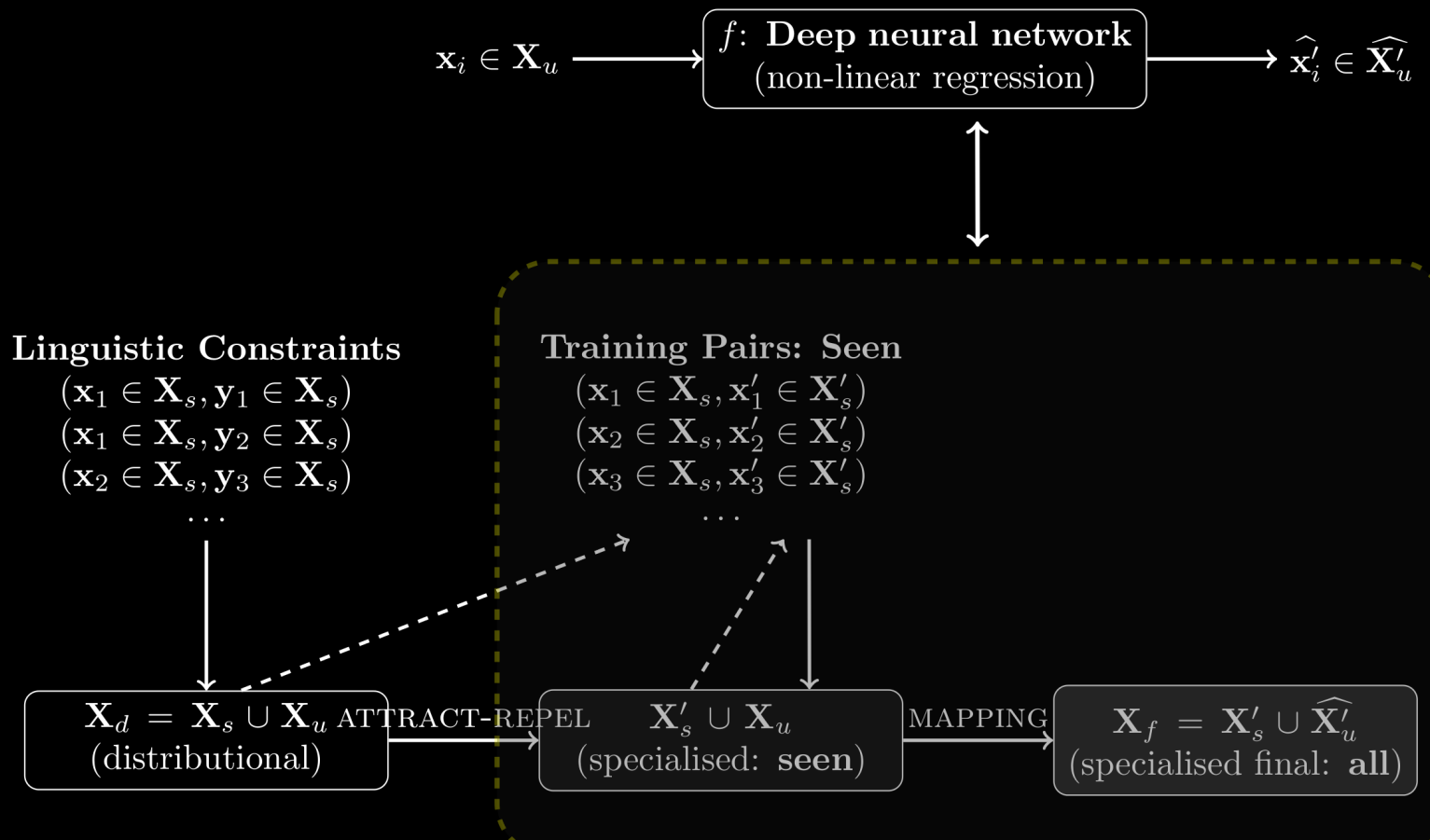
Wa want to leverage useful external knowledge also for unseen words

The main goal: specialising the full vocabulary from a distributional space

Post-Specialisation

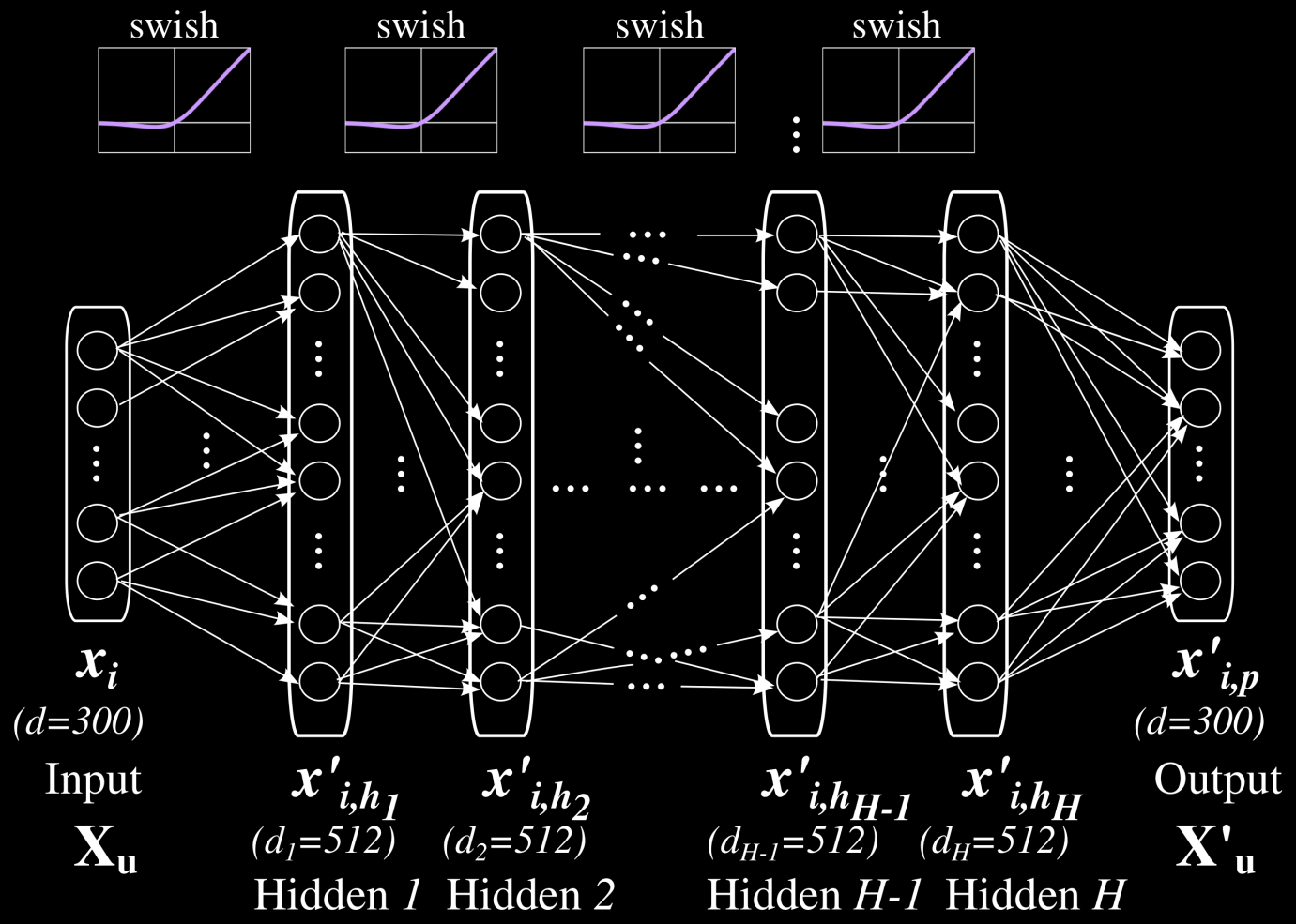
Proof-of-concept model: (Vulić et al., NAACL 2018)

A more sophisticated model: (Ponti et al., under review)



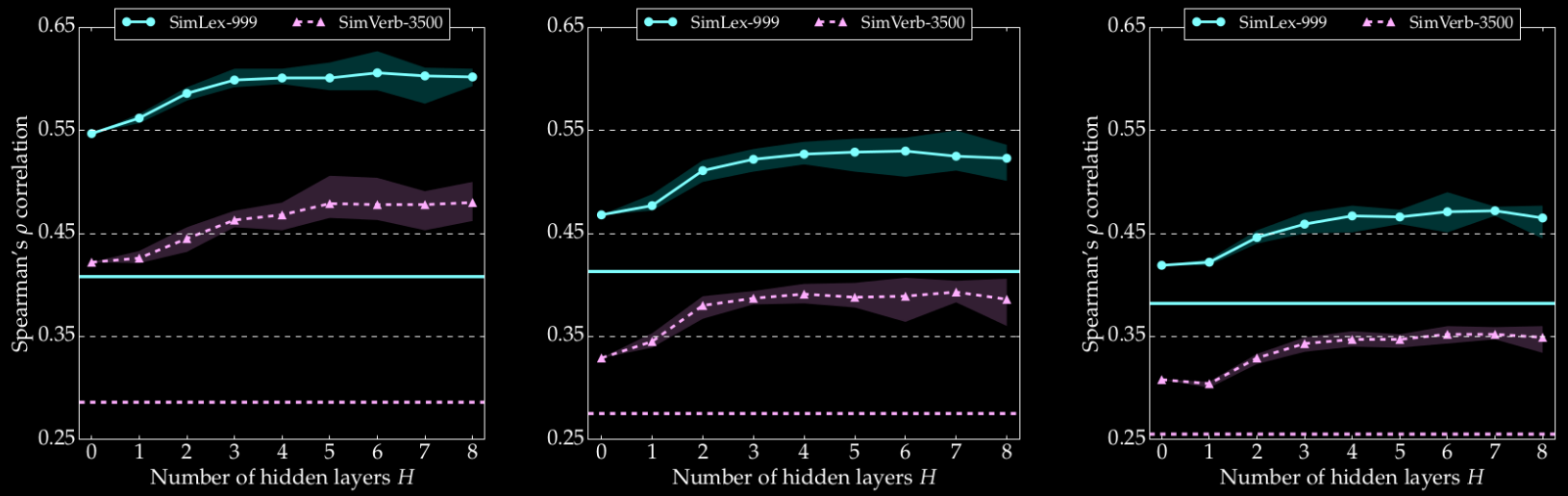
(Post-)Specialisation

Proof-of-concept work: (Vulić et al., NAACL 2018)



Post-Specialisation

	Setup: *hold-out*						Setup: *all*					
	GLOVE		SGNS-BOW2		FASTTEXT		GLOVE		SGNS-BOW2		FASTTEXT	
	SL	SV	SL	SV	SL	SV	SL	SV	SL	SV	SL	SV
Distributional: $\mathbf{X}_d$	.408	.286	.414	.275	.383	.255	.408	.286	.414	.275	.383	.255
+AR specialisation: $\mathbf{X}'_s$	.408	.286	.414	.275	.383	.255	.690	.578	.658	.544	.629	.502
++Mapping unseen: $\mathbf{X}_f$												
LINEAR-MSE	.504	.384	.447	.309	.405	.285	.690	.578	.656	.551	.628	.502
NONLINEAR-MSE	.549	.407	.484	.344	.459	.329	.694	.586	.663	.556	.631	.506
LINEAR-MM	.548	.422	.468	.329	.419	.308	.697	.582	.663	.554	.628	.487
NONLINEAR-MM	**.603**	**.480**	**.531**	**.391**	**.471**	**.349**	**.705**	**.600**	**.667**	**.562**	**.638**	**.507**



(a) GLOVE (b) SGNS-BOW2 (c) FASTTEXT

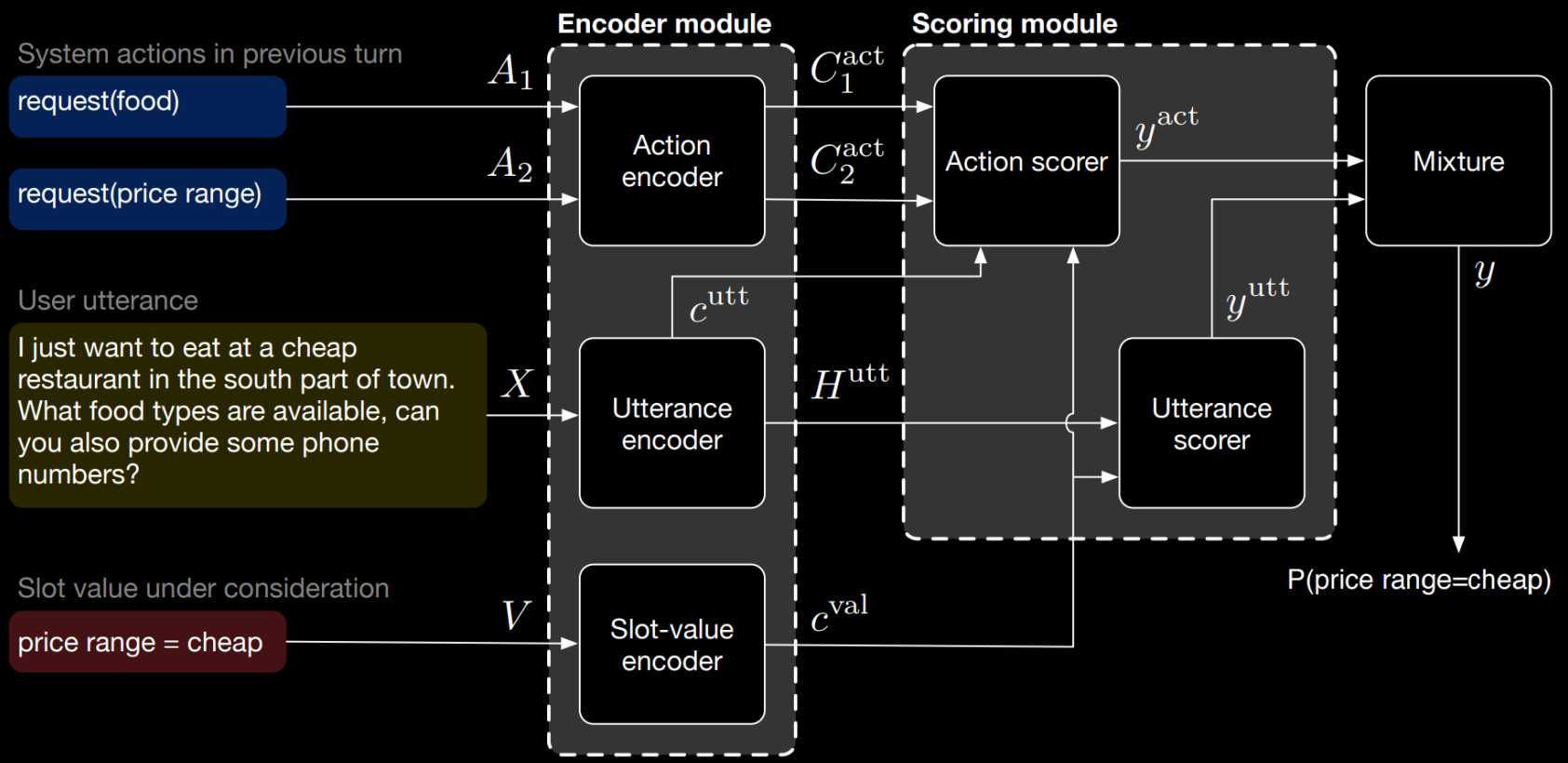
Improvements also on DST and simplification; improvements with other specialisation algorithms; improvements on DE and IT

Post-Specialisation

English	*hold-out*	*all*
Distributional: $\mathbf{X}_d$	.797	.797
+ar Spec.: $\mathbf{X}'_s \cup \mathbf{X}_u$	.797	.817
++Mapping: $\mathbf{X}_f = \mathbf{X}'_s \cup \mathbf{X}'_u$		
linear-mm	.815	.818
nonlinear-mm	**.827**	**.835**

Global-Locally Self Attentive DST (GLAD)

The same idea as NBT, but more advanced encoders



Global modules to share parameters between estimators for each slot and local modules to learn slot-specific feature representations.

That's All, Folks!