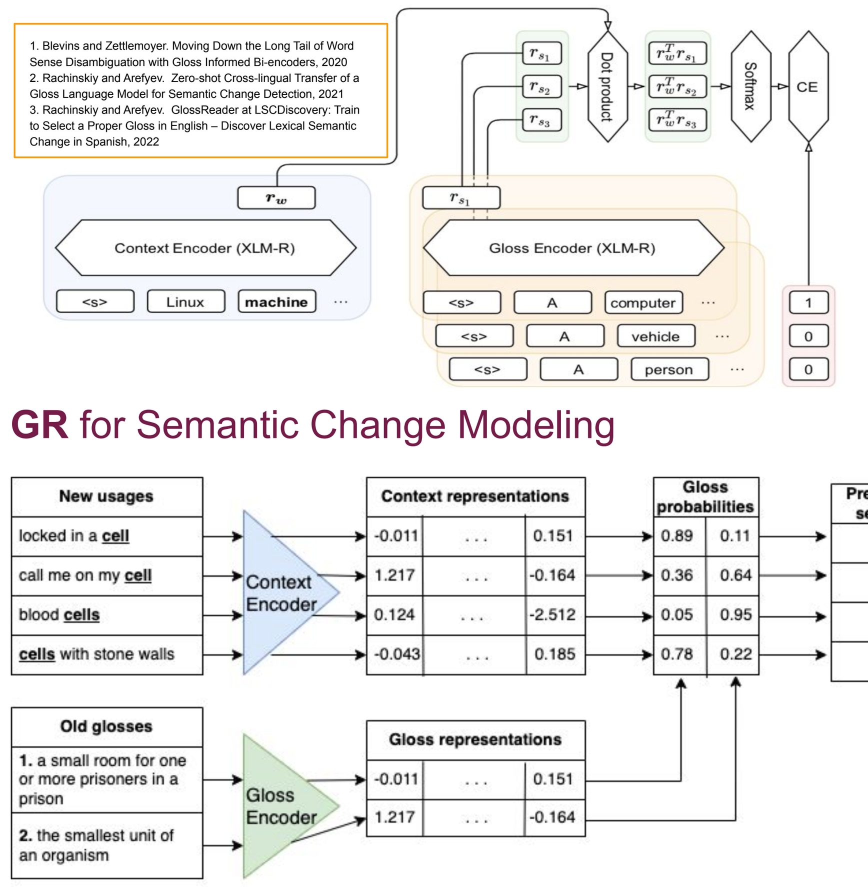
Deep-change at AXOLOTL-24: Orchestrating WSD and WSI Models for Semantic Change Modeling

Denis Kokosinskii[1,2], Mikhail Kuklin[1,3], and Nikolay Arefyev[4]
[1]Moscow State university, [2]SaluteDevices, [3]Yandex, [4]University of Oslo

WSD method

The original **GlossReader (GR)** fine-tuning



Three models finetuned on AXOLOTL-24 data:
GR FiEnRu, GR Ru, and GR Fi SG

GR fine-tuning on AXOLOTL training data

For Fi fine-tuning GR on Fi data is very important (when using as WSD system or for contextulized embeddings in AggloM)! Best when fine-tuned on 3 train sets.
For Ru fine-tuning on Ru data helps a little bit.
For De fine-tuning helps independently of the train set, but mostly on Fi with SG ?!

Method	ARI Fi	ARI Ru	ARI De	ARI FiRu	ARI AVG	F1 Fi	F1 Ru	F1 De	F1 FiRu	F1 AVG
WSD methods										
GR	0.581	0.041	0.386	0.311	0.336	0.690	◇0.721	0.694	0.706	0.702
GR FiEnRu	◇**0.649**	0.048	◇0.521	**0.348**	0.406	◇**0.756**	◇**0.750**	◇0.745	◇**0.753**	◇**0.750**
GR Ru	0.568	0.053	0.464	0.310	0.361	0.568	◇**0.750**	◇0.724	0.659	0.681
GR Fi SG	◇0.638	**0.059**	◇**0.543**	**0.348**	**0.413**	◇0.752	◇0.729	◇**0.758**	◇0.741	◇0.746
WSI methods										
Agglomerative	**0.209**	◇**0.259**	**0.316**	**0.234**	**0.261**	**0.055**	**0.152**	**0.042**	**0.104**	**0.083**
SCM methods										
AggloM	0.581	0	**0.492**	0.290	0.357	0.674	0	0.695	0.337	0.456
AggloM FiEnRu	◇0.631	0	0.485	0.315	0.372	◇0.731	0	0.639	0.366	0.457
Cluster2Sense	0.209	◇**0.259**	0.316	0.234	0.261	0.392	0.346	0.432	0.369	0.390
Outlier2Cluster $^{ru}_{fi}$	◇**0.649**	◇0.247	0.322 / 0.480	◇**0.448**	0.406 / ◇**0.459**	◇**0.756**	**0.645**	0.510 / ◇0.745	**0.701**	0.637 / ◇**0.715**

Why SCM methods get worse F1 (when not falling back to WSD)?

Intuitively, F1 of old senses should improve when we try to clean old senses from the usages of obtained senses, but
1. it is calculated for usages of old senses only, doesn't care if usages of gained senses are incorrectly put there (for most words) ⇒ only need good WSD
2. Large penalty if a single usage of some old sense is attributed to a gained sense* ⇒ trying to return anything except for the old sense labels hurts very much unless done ideally (but even then it should not help – see 1)

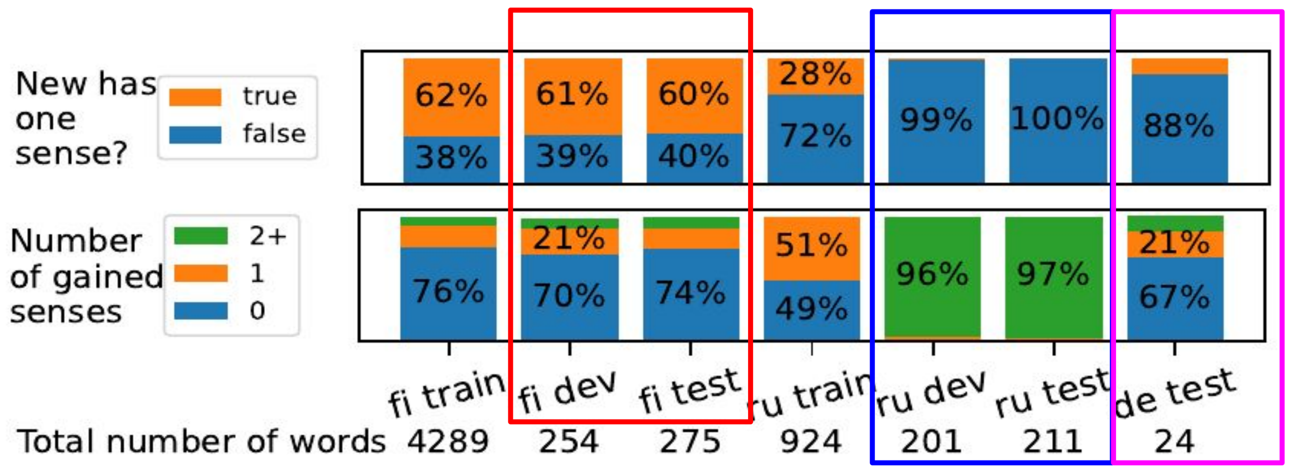
* Assume the target word has k old senses. In case when only old senses are predicted: $F = \frac{F_1+\ldots+F_k}{k}$. If we replace one of the correct predictions of sense 1 with an incorrect prediction of a gained sense: $F' = \frac{F_1'+\ldots+F_k+0}{\mathbf{k+1}} < \frac{F_1+\ldots+F_k+0}{\mathbf{k+1}}$. The drop in this metric is $\frac{F}{F'} > \frac{k+1}{k}$ E.g. in the case $k=1$, which is a frequent case in the Finnish AXOLOTL-24 dataset, an incorrect prediction of a gained sense for a single usage results in more than 2x decrease of the F1 score.

```
test_usages_predicted_senses = [
    "novel" if el not in old_senses else el
    for el in test_usages_predicted_senses
]
f1 = f1_score(
    test_usages_gold_senses,
    test_usages_predicted_senses,
    average="macro",
    zero_division=0.0,
)
```

Official eval script:
https://github.com/ltgoslo/axolotl24_shared_task/blob/main/code/evaluation/scorer_track1.py#L95

Dataset proportions

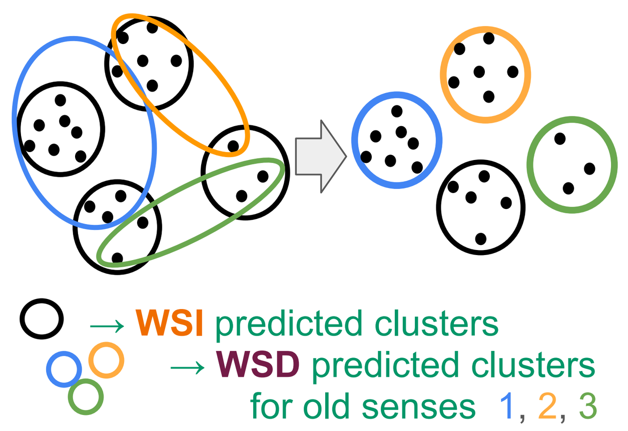
In Ru words almost always have 2 and more gained senses, so annotating with old senses is not enough – need WSI,
but in Fi ~70% words do not have gained senses (useless to do anything beyond WSD) and 60% have all new usages with 1 sense only (WSI usually gives >1 cluster ⇒ ARI=0).
In De 67% of words also do not have gained senses, but only 12% have 1 sense only ⇒ the difference in ARI is smaller than for Fi, but still large (maybe when using WSD instead of WSI sense definitions help for better grouping?)



Cluster2Sense

Inputs:
set C1 of WSI predicted clusters;
set C2 of WSD predicted clusters (corresponding to old senses);

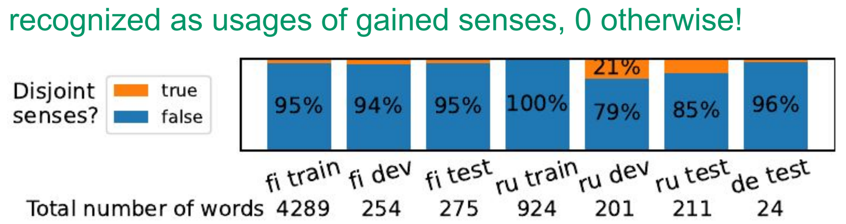
Iterative process:
1) select a pair of clusters {(c1,c2) | c1∈C1; c2∈C2} with the highest **Jaccard similarity**.
2) Relabel c1 as c2 (old sense).
3) Remove c1 from C1, c2 from C2.
Stop criteria: C1 or C2 is empty



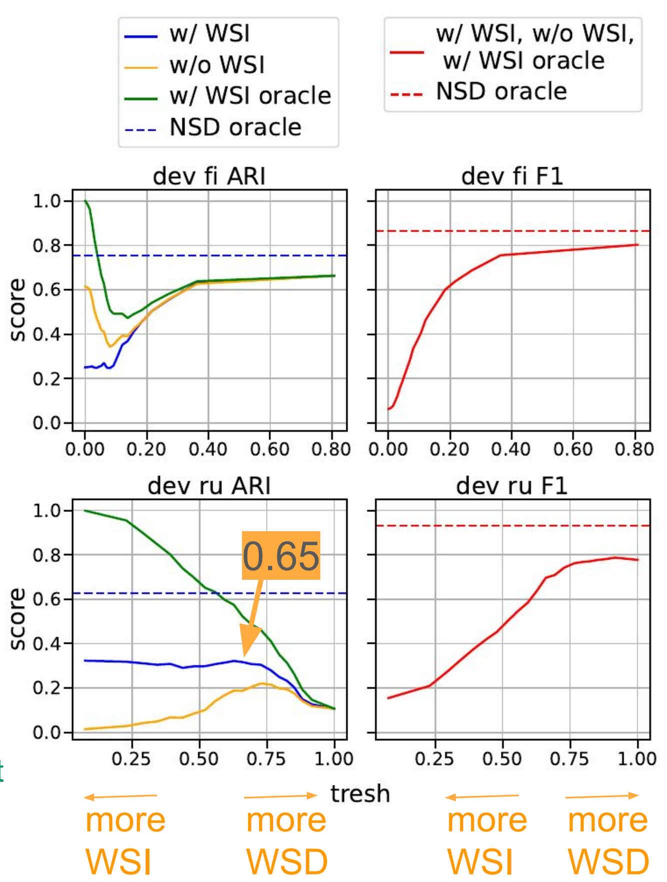
Outlier2Cluster: the threshold and oracles

Trying to clean old senses from usages of gained senses hurts F1, and also ARI on Finnish. For Russian ARI it helps!

We said even ideal NSD shouldn't help for F1, but NSD oracle improves F1?! The effect of words with no new usages of old senses: arbitrarily F1=1 when **all** usages are recognized as usages of gained senses, 0 otherwise!



WSI oracle improves ARI for Russian (>1 gained sense for 97% of words), but not Finnish (only for 7% of words). Similarly, putting all outliers to 1 cluster hurts for Russian, but not Finnish.



Results

Method	ARI Fi	ARI Ru	ARI De	ARI FiRu	ARI AVG	F1 Fi	F1 Ru	F1 De	F1 FiRu	F1 AVG
WSD methods										
GR	0.581	0.041	0.386	0.311	0.336	0.690	◇0.721	0.694	0.706	0.702
SOTA GR FiEnRu	◇**0.649**	0.048	◇0.521	**0.348**	0.406	◇**0.756**	◇**0.750**	◇0.745	◇**0.753**	◇**0.750**
GR Ru	0.568	0.053	0.464	0.310	0.361	0.568	◇**0.750**	◇0.724	0.659	0.681
SOTA GR Fi SG	◇0.638	**0.059**	◇**0.543**	**0.348**	**0.413**	◇0.752	◇0.729	◇**0.758**	◇0.741	◇0.746
WSI methods										
Agglomerative	**0.209**	◇**0.259**	**0.316**	**0.234**	**0.261**	**0.055**	**0.152**	**0.042**	**0.104**	**0.083**
SCM methods										
AggloM	0.581	0	**0.492**	0.290	0.357	0.674	0	0.695	0.337	0.456
AggloM FiEnRu	◇0.631	0	0.485	0.315	0.372	◇0.731	0	0.639	0.366	0.457
Cluster2Sense	0.209	◇**0.259**	0.316	0.234	0.261	0.392	0.346	0.432	0.369	0.390
SOTA Outlier2Cluster $^{ru}_{fi}$	◇**0.649**	◇0.247	0.322 / 0.480	◇**0.448**	0.406 / ◇**0.459**	◇**0.756**	**0.645**	0.510 / ◇0.745	**0.701**	0.637 / ◇**0.715**
Other teams										
Holotniekat	**0.596**	0.043	0.298	**0.319**	**0.312**	**0.655**	**0.661**	0.608	**0.658**	**0.641**
TartuNLP	0.437	0.098	**0.396**	0.267	0.310	0.550	0.640	0.580	0.595	0.590
IMS_Stuttgart	0.548	0	0.314	0.274	0.287	0.590	0.570	0.300	0.580	0.487
ABDN-NLP	0.553	0.009	0.102	0.281	0.221	**0.655**	0	**0.638**	0.328	0.431
WooperNLP	0.428	**0.132**	0	0.280	0.186	0.503	0.446	0	0.475	0.316
Baseline	0.023	0.079	0.022	0.051	0.041	0.230	0.260	0.130	0.245	0.207

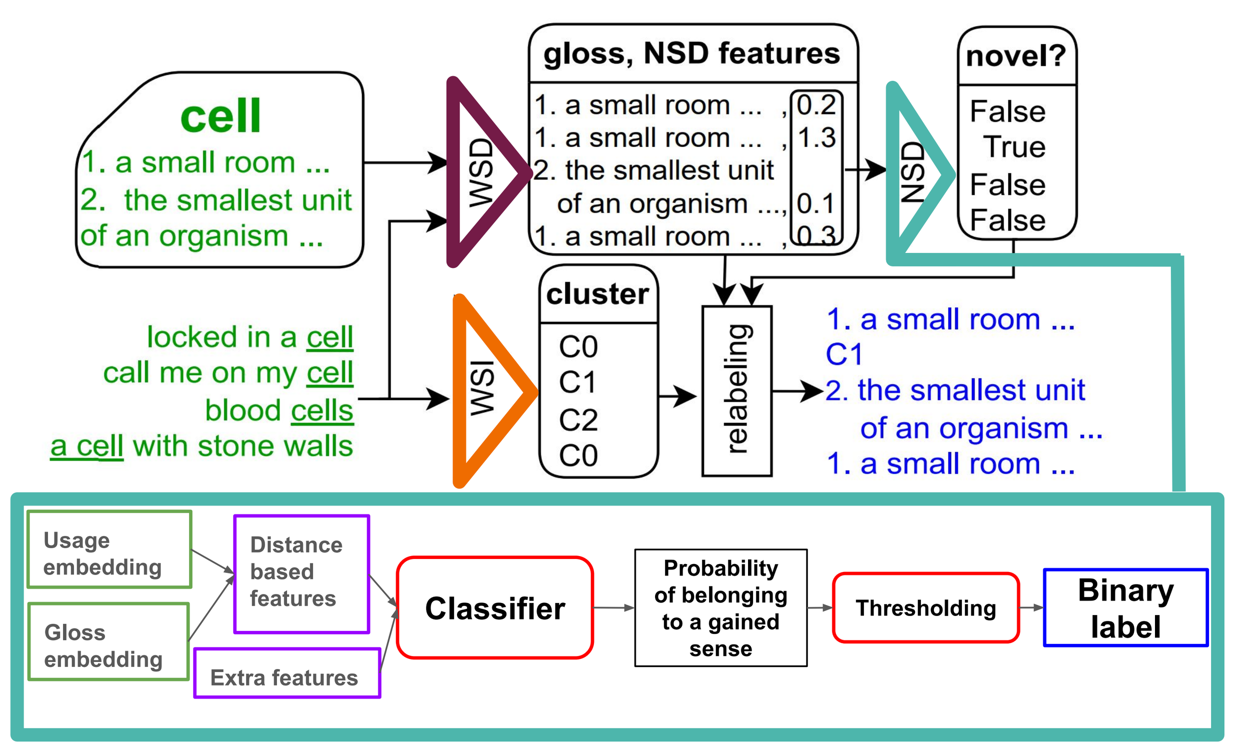
Underlined - the best of all;
Bold - the best in group;
◇ - within 5% from the best;
SOTA - outperforms all other participants according to all official leaderboard metrics (ARI and F1 averaged across all languages and FiRu);

ARI: how well new usages are grouped by their meaning?
WSI gives much better grouping than WSD for Ru, but much worse for Fi, and a bit worse for De (there are many gained senses in Ru, few in Fi and De).
Outlier2Cluster preserves the best ARI for Fi and almost best for Ru; a small drop for De (with the Finnish NSD).
AggloM significantly improves upon Agglomerative, works surprisingly well given its simplicity (but only when old usages are available).
Cluster2Sense inherits clustering and its ARI from Agglomerative.

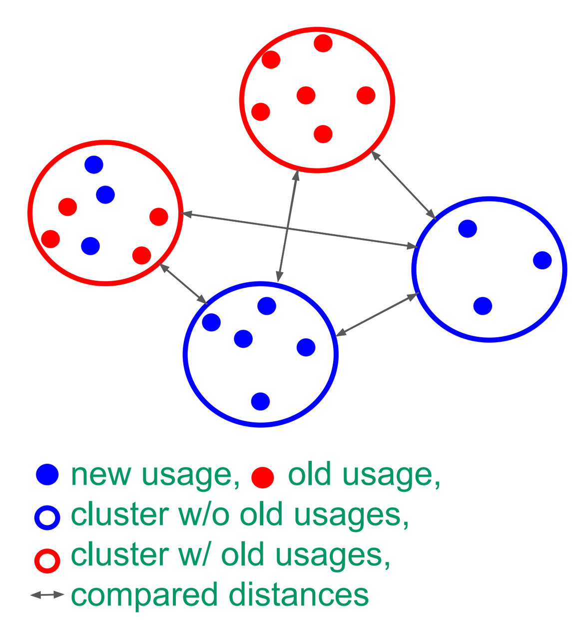
F1: how well usages of old senses are labeled with their senses?
GR FiEnRu – best WSD for usages of old senses for Fi and Ru, and De.
Outlier2Cluster preserves F1 for Fi and De (almost no positive predictions), F1 for Ru becomes significantly worse, but still comparable to the best result of other teams.
AggloM shows a bit worse results for Fi and De, comparable to the best of other teams, but cannot work for Ru.
Cluster2Sense more frequently predicts gained senses: the metric discourages it.

SCM Methods

Outlier2Cluster



AggloM

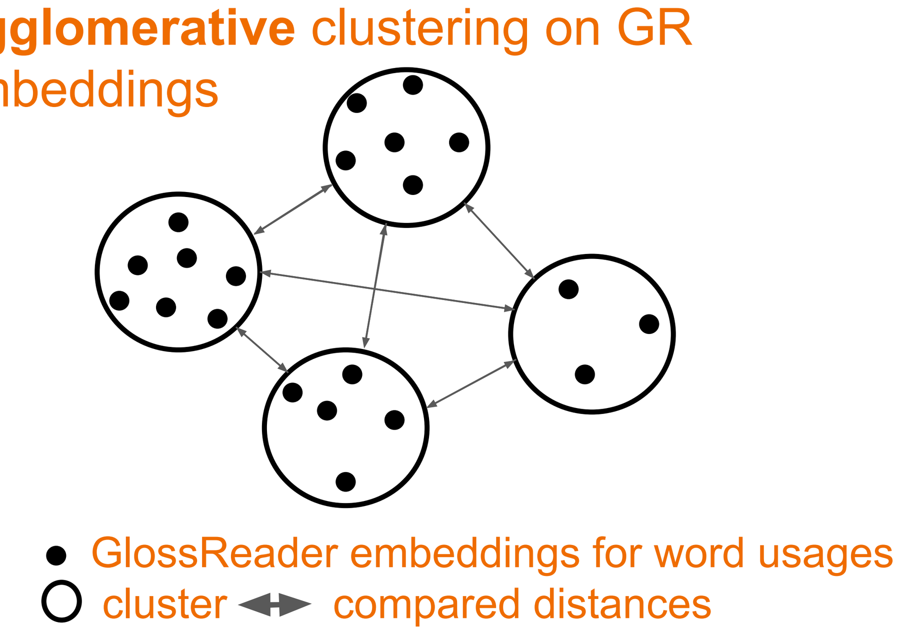


Inputs: old+new usages
Init: old usages grouped by sense, each new usage in its own cluster.
Iterative process: on each iteration the closest pair of clusters one of which does not contain old usages is merged.
Stop criteria: stop after getting **#old_senses + k** clusters

On Fi dev we selected **k=0** ⇒ each new usages ends in a cluster corresponding to an old sense. Did not use for Ru (often no old usages).

WSI method

Agglomerative clustering on GR embeddings



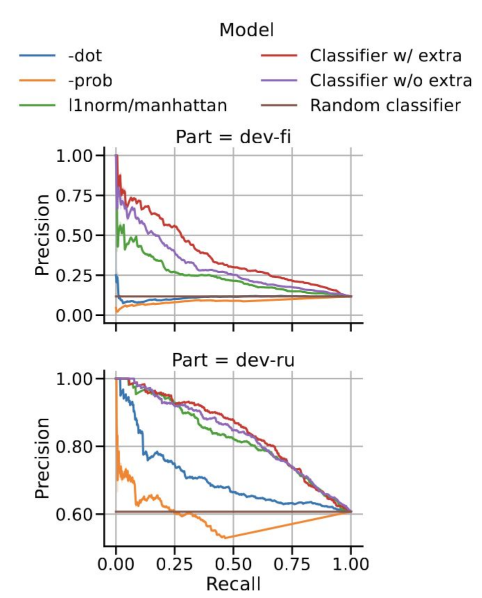
Inputs: GR embeddings of new usages, N – number of clusters
Init: each usage in its own cluster
Iterative process: on each iteration the closest pair of clusters is merged.
Stop criteria: stop after getting N clusters

NSD: Threshold Selection

Threshold of 0.65, selected on dev sets (mostly Russian)

Russian: near-optimal ARI, small loss in F1, WSI labels for ~42% of usages

Finnish and German: rarely returns WSI labels (<1%), almost like pure WSD



NSD model ablation study

Model	dev fi AP GR	dev fi AP GR FiEnRu	dev ru AP GR	dev ru AP GR FiEnRu
single features				
cosine	0.106	0.110	0.685	0.695
euclid.	0.106	0.110	0.684	0.694
l2/euclid.	0.106	0.110	0.685	0.695
manh.	0.106	0.113	0.685	0.690
l1/manh.	**0.154**	**0.242**	**0.816**	**0.822**
full classifiers				
classifier w/ extra	0.378		0.840	
classifier w/o extra	0.305		0.833	
best pairs of features w/o extra features				
l1/manh. + euclid.	0.194	**0.284**	0.818	**0.823**
l1/manh. + l2/euclid.	**0.195**	**0.284**	0.818	**0.823**
l1/manh. + manh.	0.192	0.277	**0.819**	**0.823**
best pairs of features w/ extra features				
l1/manh. + #old usages	0.190	**0.291**	0.820	0.827
l1/manh. + #new usages	0.153	0.249	**0.821**	**0.829**
#new usages + #old senses	**0.266**	0.266	0.643	0.643

Underlined - the best of all
Bold - the best in group

The predicted probability of the selected gloss and dot product perform much worse than a classifier. On Finnish they are comparable to a random classifier.
l1/manh. performs way better than other distances.
For the Finish dataset GlossReader provides poor embedding without fine-tuning.
Extra features improve NSD model, especially on Finnish.