Method↓ Dataset→	K&H+N			BLESS			ROOT09			EVALution		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Concat	0.909	0.906	0.904	0.811	0.812	0.811	0.636	0.675	0.646	0.531	0.544	0.525
Concat$_h$	0.983	0.984	0.983	0.891	0.889	0.889	0.712	0.721	0.716	0.57	0.573	0.571
Diff	0.888	0.886	0.885	0.801	0.803	0.802	0.627	0.655	0.638	0.521	0.531	0.528
Diff$_h$	0.941	0.942	0.941	0.861	0.859	0.860	0.683	0.692	0.686	0.536	0.54	0.539
NPB	0.713	0.604	0.55	0.759	0.756	0.755	0.788	0.789	0.788	0.53	0.537	0.503
LexNET	0.985	0.986	0.985	0.894	0.893	0.893	0.813	0.814	0.813	0.601	0.607	0.6
LexNET$_h$	0.984	0.985	0.984	0.895	0.892	0.893	0.812	0.816	0.814	0.589	0.587	0.583
NPB+Aug	-	-	0.897	-	-	0.842	-	-	0.778	-	-	0.489
LexNET+Aug	-	-	0.970	-	-	0.927	-	-	0.806	-	-	0.545
SphereRE	**0.990**	**0.989**	**0.990**	**0.938**	**0.938**	**0.938**	**0.860**	**0.862**	**0.861**	**0.62**	**0.621**	**0.62**
Improvement	-	-	**0.5%↑**	-	-	**1.1%↑**	-	-	**4.7%↑**	-	-	**2.0%↑**

Table 3: Performance comparison of lexical relation classification over four public datasets.

Features↓ Dataset→	K&H+N	BLESS	ROOT09	EVALution
w/o. SphereRE vectors	0.968	0.918	0.82	0.581
w. SphereRE vectors	0.990	0.938	0.861	0.62
Improvement	**+2.2%**	**+2.0%**	**+4.1%**	**+3.9%**

Table 4: Feature analysis in terms of F1 score.

Method↓ Relation→	SYN	ANT	HYP	MER	All
Attia et al. (2016)	0.204	0.448	0.491	0.497	0.423
Shwartz and Dagan (2016)	**0.297**	0.425	0.526	0.493	0.445
Glavas and Vulic (2018)	0.221	**0.504**	0.498	0.504	0.453
SphereRE	0.286	0.479	**0.538**	**0.539**	**0.471**

Table 5: Performance comparison over the CogALex-V shared task. (Due to space limitation, we only list the performance of top systems in CogALex-V.)

4.2.4 Feature Analysis

We further study whether adding the SphereRE vectors contributes to lexical relation classification. We remove all the these embeddings and use the rest of the features to make prediction based on the same neural architecture and parameter settings. The results are shown in Table 4. By learning the SphereRE vectors and adding them to the classifier, the performance improves in all four datasets.

4.3 Experiments over the CogALex-V Shared Task

We evaluate SphereRE over the CogALex-V shared task (Santus et al., 2016a), where participants are asked to classify 4,260 term pairs into 5 lexical relations: synonymy, antonymy, hypernymy, meronymy and random. The training set contains 3,054 pairs. This task is the most challenging because i) it considers random relations as noise, discarding it from the averaged F1 score; ii) the training set is small; and iii) it enforces lexical spilt of the training and testing sets, disabling "lexical memorization" (Levy et al., 2015).

In this shared task, GHHH (Attia et al., 2016) and LexNET (Shwartz and Dagan, 2016) are top-two systems with the highest performance. The most recent work on CogALex-V is STM (Glavas and Vulic, 2018). SphereRE achieves the averaged F1 score of 47.1% (excluding the random relations), outperforming state-of-the-art. Additionally, as reported in previous studies, the "lexical memorization" effect (Levy et al., 2015) is rather severe for hypernymy relations. Although SphereRE is fully distributional, it achieves the highest F1 score of 53.8%.

4.4 Analysis of SphereRE Vector Qualities

We conduct additional experiments to evaluate the qualities of Sphere vectors. The first set of experiments evaluates whether top-k most similar relation triples of a given relation triple share the same lexical relation type. This task is called top-k similar lexical relation retrieval. In this task, the similarity between two relation triples is quantified by the cosine similarity of the two corresponding SphereRE vectors. The score is reported by Precision@k. Higher Precision@k scores indicate SphereRE vectors with better quality, because lexical relation triples with the same lexical relation type should have similar Sphere vectors. In the experiments, we compute the Precision@k over all the labeled (training) and unlabeled (testing) sets of all five datasets. The results are shown in Table 6 in terms of Average Precision@k (AP@k) (with $k = 1, 5, 10$).

As seen, SphereRE has near perfect performance (over 95% for AP@1, over 90% for AP@5 and AP@10) over training sets of all five datasets. This is because in representation learning, all the labels (i.e., lexical relation types) of these term pairs are already known. Hence, SphereRE preserves distributional characteristics of these labeled datasets well. As for unlabeled datasets, the performance drops slightly over K&H+N, BLESS and ROOT09. The performance is not very satis-