A. SGD algorithm

We describe the algorithm to learn the ComplEx model with Stochastic Gradient Descent using only real-valued vectors.

Let us rewrite equation 11, by denoting the real part of embeddings with primes and the imaginary part with double primes: $e'_i = \mathrm{Re}(e_i)$, $e''_i = \mathrm{Im}(e_i)$, $w'_r = \mathrm{Re}(w_r)$, $w''_r = \mathrm{Im}(w_r)$. The set of parameters is $\Theta = \{e'_i, e''_i, w'_r, w''_r; \forall i \in \mathcal{E}, \forall r \in \mathcal{R}\}$, and the scoring function involves only real vectors:

$$\begin{aligned}\phi(r,s,o;\Theta) &= \langle w'_r, e'_s, e'_o\rangle + \langle w'_r, e''_s, e''_o\rangle \\ &\quad + \langle w''_r, e'_s, e''_o\rangle - \langle w''_r, e''_s, e'_o\rangle\end{aligned}$$

where each entity and each relation has two real embeddings.

Gradients are now easy to write:

$$\begin{aligned}\nabla_{e'_s}\phi(r,s,o;\Theta) &= (w'_r \odot e'_o) + (w''_r \odot e''_o)\\ \nabla_{e''_s}\phi(r,s,o;\Theta) &= (w'_r \odot e''_o) - (w''_r \odot e'_o)\\ \nabla_{e'_o}\phi(r,s,o;\Theta) &= (w'_r \odot e'_s) - (w''_r \odot e''_s)\\ \nabla_{e''_o}\phi(r,s,o;\Theta) &= (w'_r \odot e''_s) + (w''_r \odot e'_s)\\ \nabla_{w'_r}\phi(r,s,o;\Theta) &= (e'_s \odot e'_o) + (e''_s \odot e''_o)\\ \nabla_{w''_r}\phi(r,s,o;\Theta) &= (e'_s \odot e''_o) - (e''_s \odot e'_o)\end{aligned}$$

where $\odot$ is the element-wise (Hadamard) product.

As stated in equation 8 we use the sigmoid link function, and minimize the L^2-regularized negative log-likelihood:

$$\begin{aligned}\gamma(\Omega;\Theta) &= \sum_{r(s,o)\in\Omega} \log(1+\exp(-\mathbf{Y}_{rso}\phi(s,r,o;\Theta))) \\ &\quad +\lambda||\Theta||_2^2 \ .\end{aligned}$$

To handle regularization, note that the squared L^2-norm of a complex vector $v = v' + iv''$ is the sum of the squared modulus of each entry:

$$\begin{aligned}||v||_2^2 &= \sum_j \sqrt{v'^2_j + v''^2_j}^2 \\ &= \sum_j v'^2_j + \sum_j v''^2_j \\ &= ||v'||_2^2 + ||v''||_2^2\end{aligned}$$

which is actually the sum of the L^2-norms of the vectors of the real and imaginary parts.

Algorithm 1 SGD for the ComplEx model

input Training set Ω, Validation set Ω_v, learning rate α, embedding dim. k, regularization factor λ, negative ratio η, batch size b, max iter m, early stopping s.
  $e'_i \leftarrow \mathrm{randn}(k)$, $e''_i \leftarrow \mathrm{randn}(k)$ for each $i \in \mathcal{E}$
  $w'_i \leftarrow \mathrm{randn}(k)$, $w''_i \leftarrow \mathrm{randn}(k)$ for each $i \in \mathcal{R}$
  for $i = 1, \cdots, m$ **do**
    for $j = 1..|\Omega|/b$ **do**
      $\Omega_b \leftarrow \mathrm{sample}(\Omega, b, \eta)$
      Update embeddings w.r.t.:
        $\sum_{r(s,o)\in\Omega_b} \nabla\gamma(\{r(s,o)\};\Theta)$
      Update learning rate α using Adagrad
    end for
    if $i \mod s = 0$ **then**
      break if filteredMRR or AP on Ω_v decreased
    end if
  end for

We can finally write the gradient of γ with respect to a *real* embedding v for one triple $r(s,o)$:

$$\begin{aligned}\nabla_v\gamma(\{r(s,o)\};\Theta) &= -\mathbf{Y}_{rso}\phi(s,r,o;\Theta)\sigma(\nabla_v\phi(r,s,o;\Theta)) \\ &\quad +2\lambda v\end{aligned}$$

where $\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}}$ is the sigmoid function.

Algorithm 1 describes SGD for this formulation of the scoring function. When Ω contains only positive triples, we generate η negatives per positive train triple, by corrupting either the subject or the object of the positive triple, as described in Bordes et al. (2013b).

B. WN18 embeddings visualization

We used principal component analysis (PCA) to visualize embeddings of the relations of the wordnet dataset (WN18). We plotted the four first components of the best DistMult and ComplEx model's embeddings in Figure 4. For the ComplEx model, we simply concatenated the real and imaginary parts of each embedding.

Most of WN18 relations describe hierarchies, and are thus antisymmetric. Each of these hierarchic relations has its inverse relation in the dataset. For example: `hypernym` / `hyponym`, `part_of` / `has_part`, `synset_domain_topic_of` / `member_of_domain_topic`. Since DistMult is unable to model antisymmetry, it will correctly represent the nature of each pair of opposite relations, but not the direction of the relations. Loosely speaking, in the `hypernym` / `hyponym` pair the nature is sharing semantics, and the direction is that one entity generalizes the semantics of the other. This makes DistMult reprensenting the opposite

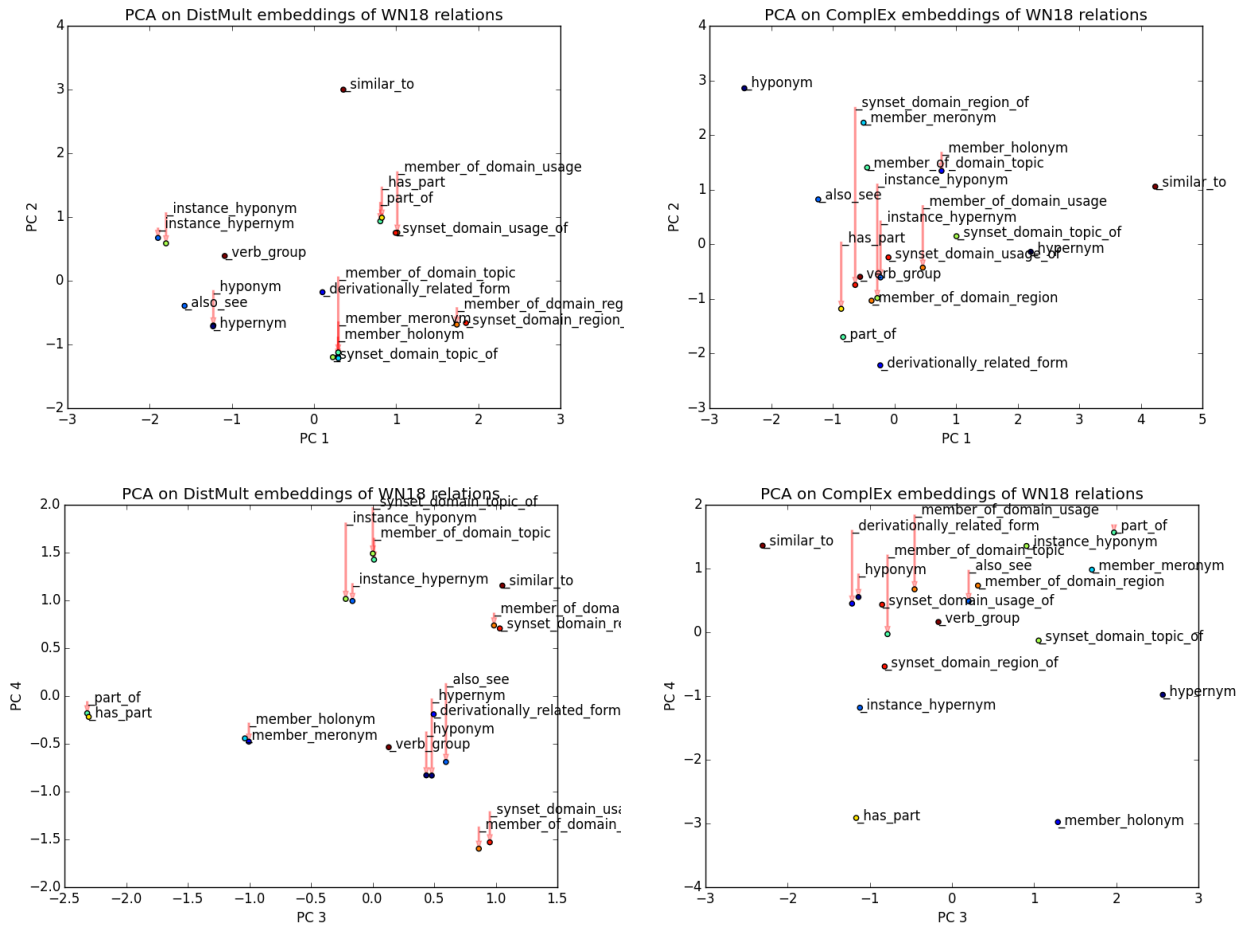

Figure 4. Plots of the first and second (Top), third and fourth (Bottom) components of the WN18 relations embeddings using PCA. Left: DistMult embeddings. Right: ComplEx embeddings. Opposite relations are clustered together by DistMult while correctly separated by ComplEx.

relations with very close embeddings, as Figure 4 shows. It is especially striking for the third and fourth principal component (bottom-left). Conversely, ComplEx manages to oppose spatially the opposite relations.