The main trick that I used was for ranking.

Basically, if sentences is (a,b,c,d,?,e,f,g,....), and you have words to rank w1,w2,....wk, then, rank ith word according to frequency of bigram, that is, define score for each wi as ***count(d,wi)+count(e,wi)*** where counts are coming from train set. Then sort in descending order. This gave me overall jump.

In my final model, I did ranking using this, and ties were broken my embeddings based model, which was trained using gensim, and ranking was done using gensim's internal function ***`predict_output_word.`*** `Without ranking defined above, mrr was ~0.3`