1. Brill's Tagger and Tagger Evaluation

	textcz2.ptg.txt	**texten2.ptg.txt**
T = rest, S = last 40, H = pre-last 20	0.752112227004623	0.8906213334028629
T = rest, S = first 40, H = after-first 20	0.7729422392263138	0.8878723903354445
T = rest, S = first 40, H = last 20	0.7811785960996865	0.8894362342638205
T = rest, S = last 40, H = first 20	0.7587013125033211	0.8921336010220843
T = rest, S = after-first 40, H = first 20	0.7779842820730671	0.8957301496272353
Mean accuracy	0.7685837313814023	0.8911587417302895
Standard deviation of accuracy	0.011269163416409511	0.002679703937821701

2. Unsupervised Learning: HMM Tagging

		textcz2.ptg.txt	**texten2.ptg.txt**
Brill`s tagger	Full tag	0.7685837313814023	0.8911587417302895
Supervised learning	Full tag	0.8048780487804879 (30294 out of 37638)	0.9286105389408912 (35615 out of 38353)
	Short tag	0.8548541367766619 (32175 out of 37638)	
Unsupervised learning	Short tag	0.7095754290876242 (26707 out of 37638) (result of 3th iteration)	0.8223867754804057 31541 out of 38353 (result of 2nd iteration)

Accuracy is lower for Czech than for English: it might be because of many unseen words (in our case with small probability) and bigger number of tags to assign. All these is due to the rich morphology of Czech language.

Brill`s tagger shows lower accuracy than supervised HMM tagger and it is language-dependent (difference is probably the same as described above) and data-dependent (considering its quality). I predict that we would encounter the same behavior for HMM tagger regarding the data if we conducted the same experiment as for Brill`s tagger. The main difference is about the algorithm used in these two taggers. HMM tagger uses much more sophisticated algorithm (HMM, different probabilities and smoothing) which allows to achieve better performance.

Unsupervised HMM tagger shows lower quality in comparison to both Brill`s and supervised HMM taggers. It is not surprising because usually unsupervised learning needs a lot of data to achieve any good results. The amount of data we have is not enough to handle the task of tagging perfectly because many patterns cannot be inferred because they are not frequent in the data.

Problems: Unsupervised tagging is really time-consuming. To avoid long computations, I used numpy matrices which is not really memory efficient but I had to sacrifice memory to speed. That is why, I could not initialize big matrices. Thus, the main problem was Baum-Welch for Czech. I could not initialize matrices neither for full set of tags nor for trigram model (and obviously not for both). Instead, I used only first two symbols of a tag (the resulting amount of tags is the only one that could be initialized) and bigram language model. As a results, we have kind of unequal comparison between (Brill`s and supervised HMM) and unsupervised HMM because they predict different tags for Czech. However, if we consider that I actually reduced number of possible tags for Czech (which should have increased the accuracy because now tags easier to predict), we will understand that prediction of full tags for Czech by unsupervised HMM would have even lower accuracy. Thus, we can conclude that unsupervised tagging is the worst regarding accuracy.

After several iterations, results for both languages started getting worse (maybe the threshold for convergence was chosen too small and the model overfitted). Thus, I did not wait for the actual convergence and interrupted training at the peak of accuracy.