

# GenSen

## Qu'est ce que GenSen

GenSen est une technique pour apprendre des objectifs, des représentations de phrases en utilisant un apprentissage multitâche.

## En pratique

Les jeux de données utilisés ici sont SUBJ, TREC et DBpedia.

Concernant la préparation de l'entraînement multitâches :

Nous avons besoin d'un certain nombre de tâches, d'un encodeur partagé pour toutes les tâches et un nombre de décodeurs (du même nombre que celui des tâches).

Maintenant, les prochaines étapes vont boucler :

Nous récupérons une tâche  $i$  et son jeu de données associé pour créer des paires de valeurs entrée/sortie.

La valeur d'entrée va alors être encodée pour créer une représentation qui va ensuite passer dans le décodeur associé à la tâche, pour générer une prédiction.

Le modèle utilisé est comme suit :

L'encodeur partagé utilise une table de recherche de mots communs et GRU (Gated Recurrent Unit) qui est en quelque sorte un réseau de neurones artificiel qui peut non seulement utiliser des données simples mais également des séquences de données entières comme de la vidéo.

## Résultats

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	SICK-R	SICK-E	STSB	$\Delta$
<i>Transfer approaches</i>											
FastSent	70.8	78.4	88.7	80.6	-	76.8	72.2/80.3	-	-	-	-
FastSent+AE	71.8	76.7	88.8	81.5	-	80.4	71.2/79.1	-	-	-	-
NMT En-Fr	64.7	70.1	84.9	81.5	-	82.8	-	-	-	-	-
CNN-LSTM	77.8	82.1	93.6	89.4	-	92.6	76.5/83.8	0.862	-	-	-
Skipthought	76.5	80.1	93.6	87.1	82.0	92.2	73.0/82.0	0.858	82.3	-	-
Skipthought + LN	79.4	83.1	93.7	89.3	82.9	88.4	-	0.858	79.5	72.1/70.2	-
Word Embedding Average	-	-	-	-	82.2	-	-	0.860	84.6	-	-
DiscSent + BiGRU	-	-	88.6	-	-	81.0	71.6/-	-	-	-	-
DiscSent + unigram	-	-	92.7	-	-	87.9	72.5/-	-	-	-	-
DiscSent + embed	-	-	93.0	-	-	87.2	75.0/-	-	-	-	-
Byte mLSTM	<b>86.9</b>	<b>91.4</b>	<b>94.6</b>	88.5	-	-	75.0/82.8	0.792	-	-	-
Infersent (SST)	(*)	83.7	90.2	89.5	(*)	86.0	72.7/80.9	0.863	83.1	-	-
Infersent (SNLI)	79.9	84.6	92.1	89.8	83.3	88.7	75.1/82.3	<u>0.885</u>	<u>86.3</u>	-	-
Infersent (AllNLI)	81.1	86.3	92.4	<u>90.2</u>	<b>84.6</b>	88.2	76.2/83.1	0.884	<u>86.3</u>	<u>75.8/75.5</u>	0.0
<i>Our Models</i>											
+STN	78.9	85.8	93.7	87.2	80.4	84.2	72.4/81.6	0.840	82.1	72.9/72.4	-2.56
+STN +Fr +De	80.3	85.1	93.5	90.1	83.3	92.6	77.1/83.3	0.864	84.8	77.1/77.1	0.01
+STN +Fr +De +NLI	81.2	86.4	93.4	90.8	84.0	93.2	76.6/82.7	0.884	87.0	79.2/79.1	0.99
+STN +Fr +De +NLI +L	81.7	87.3	<u>94.2</u>	90.8	84.0	<b>94.2</b>	77.1/83.0	0.887	87.1	78.7/78.2	1.33
+STN +Fr +De +NLI +L +STP	82.7	88.0	94.1	91.2	<u>84.5</u>	92.4	77.8/83.9	0.885	86.8	78.7/78.4	1.44
+STN +Fr +De +NLI +2L +STP	<u>82.8</u>	<u>88.3</u>	94.0	<b>91.3</b>	83.6	92.6	77.4/83.3	0.884	87.6	<b>79.2/79.1</b>	1.47
+STN +Fr +De +NLI +L +STP +Par	82.5	87.7	94.0	90.9	83.2	93.0	<b>78.6/84.4</b>	<b>0.888</b>	<b>87.8</b>	78.9/78.6	<b>1.48</b>
<i>Approaches trained from scratch on these tasks</i>											
Naive Bayes SVM	79.4	81.8	93.2	86.3	83.1	-	-	-	-	-	-
AdaSent	83.1	86.3	95.5	93.3	-	92.4	-	-	-	-	-
TF-KLD	-	-	-	-	-	-	80.4/85.9	-	-	-	-
Illinois LH	-	-	-	-	-	-	-	-	84.5	-	-
Dependency tree LSTM	-	-	-	-	-	-	-	0.868	-	-	-
Neural Semantic Encoder	-	-	-	-	89.7	-	-	-	-	-	-
BLSTM-2DCNN	82.3	-	94.0	-	89.5	96.1	-	-	-	-	-

Ici, nous avons les résultats donnés à chaque tâche indépendamment, chaque colonne étant une tâche utilisant un modèle linéaire entraîné grâce aux représentations de chaque modèle.

Ci-dessus, STN, Fr, De, NLI, L, STP et Par correspondent respectivement à “skip-thought next”, traduction en français, traduction en allemand, inférence de langage naturel, modèle large, modèle large de 2 couches, “skip-thought previous” et parsing.