Support des langues étrangères

(à l'Anglais)

© Mathieu Rigard, pour la traduction française, m.rigard@utopia-french.tech https://utopia-french.tech
avec l'autorisation de l'Auteur.
© Bruce Wilcox, mailto: gowilcox@gmail.com www.brilligunderstanding.com
Révision 11/04/2017 cs7.61

Besoin : version 1.0, à relire et corriger.

Langue étrangère : Présentation

ChatScript vient nativement avec un support complet de l'anglais. Si vous voulez utiliser une langue différente, vous avez besoin d'une variété de choses :

- Le support du pos-tagging (facultatif si vous n'utilisez pas de mots-clés POS, le POS). *[Le POS (Part Of Speech) Tagging consiste en le découpage de l'entrée de l'utilisateur en parties faisant sens comme sujet, verbe, complément. Note du traducteur]*

- Le support de vérification orthographique

- Des concepts dans la langue (que vous utilisez)
- Des fichiers de substitutions LIVEDATA appropriées à la langue
- Des Modèles dans la langue
- Des Sortie dans la langue
- Des Lemmes (avec des pos-taggers étrangers ou accessible depuis ^pos(canonical)) *[un Lemme est unes unité autonome constituante du lexique d'une langue Note du traducteur]*

ChatScript propose un paramètre pour la ligne de commande language= qui indique à CS la langue que vous souhaitez utiliser. Il est par défaut en ANGLAIS. Les effets de ce paramètre sont les suivants :
- Si ce n'est pas en ANGLAIS, le pos-tagging interne (autre que le marquage des balises anglaises et des numéros et dates possibles) et l'analyse syntaxique sont désactivés.
- Si le treetagger est sous licence et possède cette langue, il pourra tagger le POS.
- Le système utilisera DICT/language .
- Le système utilisera LIVEDATA/language

- Le compilateur de script compilera automatiquement les lignes marquées pour être compilées conditionnellement avec la langue (voir les commentaires sur la langue).
- Le système stockera également le fichier de sujet de l'utilisateur avec le suffixe de langue.

ENTRÉES et SORTIES

ChatScript prend en charge l'UTF8, ce qui vous laisse libre de la Sortie ou des Modèles dans votre langue. Idem pour LIVEDATA.

ChatScript prend en charge deux types de commentaires de compilation conditionnelle. Les commentaires sur une ligne ressemblent à ceci:
#ENGLISH cette ligne va compiler si la langue est l'anglais mais pas si la langue est l'Allemand.
#GERMAN cette ligne ne compilera pas si la langue est l'anglais mais compilera si la langue est l'allemand.

Comme toujours, ces commentaires vont jusqu'à la fin de la ligne. L'autre commentaire est le commentaire de bloc comme ceci:

##<<ENGLISH ces lignes seront compilées en anglais
jusqu'à un commentaire de bloc de fermeture normal ##>>

À l'aide de la compilation conditionnelle, vous pouvez faire en sorte que l'anglais et d'autres versions du code soient installées côte à côte si vous le souhaitez.

DICTIONNAIRE

Le fichier de dictionnaire peut être juste une liste de mots de la langue, un par ligne. Vous devez lister toutes les conjugaisons d'un mot car il n'y a pas de support intégré pour le comprendre. Vous pouvez également ajouter des balises POS tag équivalentes anglaises (voir les exemples dans les dictionnaires de langues étrangères existants) si vous souhaitez utiliser des mots-clés existants liés à des balises pos tag.

En plus des mots normaux, il y a un fichier LIVEDATA /.../ numbers.txt qui, pour une langue, décrit un nombre en toutes lettres et sa signification en chiffre.

`:buildforeign language` peut être utilisée pour reconstruire un dictionnaire étranger étant donné les données rawwords dans le répertoire TreeTagger (que vous n'avez pas) et

POS-TAGS ET LEMMAS

Si vous voulez des valeurs POS réelles et des lemmes (forme canonique d'un mot), vous aurez besoin d'un POS-tagger ou d’utiliser ^pos(canonical) sur un mot. Bien qu'il soit possible de raccorder un tagueur externe via un appel Web, il sera nettement plus lent qu'un système intégré.

ChatScript prend en charge le système TreeTagger intégré, qui prend en charge un certain nombre de langues. Cependant, vous ne pouvez l'utiliser que si vous n’en possédez pas une licence. Vous pouvez l'essayer en utilisant ^popen, comme dans le bot allemand, mais il sera lent car il doit réinitialiser TreeTagger pour chaque phrase. Le système intégré ne le fait pas. Une licence (par langue) est d'environ 1000 $ pour une utilisation universelle à vie. Vous pouvez me contacter si vous voulez la prendre.

Exécution de CS dans une langue étrangère

Pour l'exécuter dans une langue, utilisez le paramètre language= en ligne de commande. Pour toute instance CS, vous ne pouvez prendre en charge qu'une seule langue à la fois (car CS ne peut pas charger plusieurs dictionnaires ou détecter automatiquement une langue). Donc, si vous voulez avoir des serveurs pour plusieurs langues, vous avez besoin de plusieurs serveurs, chacun étant redirigé de manière appropriée.

Le fichier de l'utilisateur où se trouvent les Topic est nommé par username-botname-language (pour l’anglais oubliez le -language). Cela signifie qu'un utilisateur discutant avec un serveur CS allemand a une conversation indépendante de n'importe quelle conversation qu'il a eue avec le serveur anglais. Si vous utilisez des fichiers LTM pour stocker des données à long terme sur l'utilisateur, vous pouvez choisir si ces données sont uniques par langue ou partagées, puisque le nom du fichier est écrit dans un script.

Traduire des concepts

Il y a un code intégré pour traduire les concepts en utilisant Google Traduction. Cela nécessite que vous ayez une clé d'api pour Google Traduction (mais vous pouvez vous inscrire gratuitement et obtenir un crédit de 300$ valable pendant 3 mois, ce qui est suffisant pour faire tout votre travail de traduction probablement). Vous dites à CS ceci comme un paramètre de ligne de commande:

apikey=AIzaSyAxxxx

Quand je veux traduire tous les concepts de niveau 0, je fais ce qui suit:

1. effacer le contenu du dossier TOPIC
2. :build 0
3. exécuter CS en utilisant le paramètre de ligne de commande noboot et votre apikey
4. :sortconcept x
5. :translateconcept french monfichier

Si vous lancez ^csboot et que cela génère de nouvelles données conceptuelles, vous avez besoin de noboot, sinon cela n'a pas d'importance.

:sortconcept x localise tous les concepts actuellement définis (donc seulement :build 0) et les écrit dans un fichier de niveau supérieur nommé concepts.top avec un concept par ligne. Ce fichier sera lu à l'étape suivante.

:translateconcept utilise l'apikey. Il lit chaque ligne de concepts.top (1 ligne par concept) et appelle google translate pour la langue que vous avez nommée, en enregistrant les résultats dans le chemin/fichier que vous avez donné. Actuellement, cela ne reconnaît que les noms de langue suivants: german, french, italian, spanish, russian, hindi (allemand, français, italien, espagnol, russe, hindi). Je pourrais en ajouter plus si nécessaire.

Le fichier résultant ajoutera automatiquement chaque ligne avec des marqueurs de compilation conditionnels pour la langue que vous avez nommée, de sorte que vous pouvez directement l'ajouter à votre bot et il ne compilera que lorsque vous êtes dans ce mode de langue.

Si vous voulez traduire des concepts de votre bot, alors faites ce qui suit:

1. effacer le contenu du dossier TOPIC
2. :build Harry (ou quel que soit le nom de votre bot)
3. exécuter CS en utilisant le paramètre de ligne de commande noboot et votre apikey
4. :sortconcept x
5. :translateconcept french myfilename

Si vous voulez juste traduire un seul concept / Topic, vous pouvez appeler

:translateconcept ~myconcept monfichier french

Il fournira, en tant que sous-produit, la forme anglaise du concept triée sur une seule ligne dans cset.txt. Si vous ne donnez pas une langue et un nom de fichier, alors il va juste trier votre concept anglais et l'écrire.