

Rapport ALAO

Les verbes de mouvement dans l'enseignement du russe

Sommaire

I- <u>Une présentation du projet</u>	p.1
Problématique		
Description des verbes de mouvement en russe		
II- <u>Des exercices pour travailler les verbes de mouvement</u>		
⇒ Pour l'exercice 1		
Script (Main.java, create_corpus.py)	p.2
⇒ Pour l'exercice 2		
Script (Main.java)		
⇒ Pour l'exercice 3		
Script (Main.java, Sample.java, request_ruscorpora.py)	p.3
III- <u>Les difficultés rencontrées</u>		
⇒ Pour l'exercice 1		
⇒ Pour l'exercice 3	p.4
IV- <u>Une prise de recul</u>	p.5

I- Une présentation du projet

Ce projet est une requête formulée par une des enseignantes de la langue russe à l'université de Grenoble.

Une des difficultés et spécificités de la langue russe repose sur les verbes de mouvement. C'est une notion assez complexe, sans équivalence avec le français qui nécessite d'être travaillée progressivement c'est-à-dire dès le niveau A2. Notre projet s'inscrit pour un travail plus avancé et pour un public de niveau B1.

Problématique

Les verbes de mouvement en russe se construisent à partir d'un préfixe que l'on ajoute au radical du verbe. On dit qu'ils sont préverbés, ils perdent leurs caractéristiques de mouvement simple pour se transformer en couple aspectuel (perfectif/imperfectif). Seuls quelques couples verbaux sont concernés.

Description des verbes de mouvement en russe : Pour un verbe français, il y a deux verbes en russe, un perfectif qui servira à exprimer le passé ou le futur pour une action brève ou terminée et un verbe imperfectif qui servira à exprimer le présent mais également le passé ou le futur pour une action qui s'inscrit sur la durée ou bien en train de se faire. Pour ce verbe imperfectif, il existe deux formes, une forme déterminée et une indéterminée

Par exemple, pour le verbe "aller à pied" :

- идти est sa forme déterminée et décrit une action ponctuelle
- ходить est sa forme indéterminée et décrit un processus

Notre travail consiste ici à offrir une série d'exercices de difficulté croissante qui aideront l'apprenant à maîtriser ou renforcer ses connaissances dans l'utilisation des préfixes pour les verbes russes de mouvement.

II- Des exercices pour travailler les verbes de mouvement

Notre outil permettra d'analyser les erreurs les plus courantes dans la production de phrases avec ce type de verbes. Nous proposons trois types d'exercices de complexité différente et chacun est conçu en langage Java.

- exercice 1 = aide à comprendre certaines difficultés avec la signification des préfixes.
- exercice 2 = l'utilisateur s'exerce à la création correcte du verbe avec préfixe. Il existe des règles spéciales pour ajouter les préfixes au verbe.
- exercice 3 = on demande à l'utilisateur de traduire une phrase du français vers le russe.

Pour chaque exercice, nous proposons un algorithme qui donne une réponse vrai ou faux. Pour le troisième exercice, l'algorithme analyse la production écrite de l'utilisateur selon trois critères: lemme, temps et aspect.

⇒ **Pour l'exercice 1**, il s'agit de choisir un verbe correct en russe pour le verbe en français. Exemple:

entrer 1.входить
2.выходить
3.приходить
4.уходить

Globalement, il y a une variante de 400 verbes de mouvements, il y a en effet 14 formes radicales donc 28 si on reprend les formes perfectifs et imperfectifs et pour chacune de ces formes on peut ajouter 14 formes de préfixes.

Script (Main.java, create_corpus.py)

L'algorithme conçu pour cet exercice prend par hasard le couple clé-valeur où la clé est un verbe français et la valeur est un verbe russe. Il propose la clé (verbe français) à l'utilisateur et vérifie que sa réponse corresponde à la valeur (verbe russe). Nous utilisons le dictionnaire des verbes créés auparavant (verbs_corpus.txt), donc à chaque fois l'utilisateur peut avoir différents verbes à traduire.

Le fichier create_corpus.py crée un corpus de verbes de mouvements russes à partir de 7 paires de verbes de base. L'idée générale de ce corpus est de fournir à l'utilisateur différentes formes de verbes basées sur ces racines. Ainsi, il pourra exploiter non seulement les verbes de base mais aussi travailler sur d'autres verbes avec différents suffixes et préfixes. Comme il n'y a pas de tel corpus nous avons utilisé une liste de 14 verbes de base puis nous avons trouvé tous les mots de la même racine en interrogeant le site <https://wordroot.ru/>. Après cela, nous avons trié le résultat par terminaisons de verbes et obtenu une liste étendue de verbes avec la même racine. Nous avons ajouté une vérification supplémentaire sur <https://ru.wiktionary.org/wiki/>. Nous vérifions si le verbe de la liste a un attribut de mouvement dans le wiktionary, et seulement après cela, nous l'ajoutons dans la liste finale (fichier verbs_list.txt). Ensuite, grâce au module importé de Google translate nous traduisons tous ces verbes et nous obtenons le corpus de verbes en deux langues qui sera utilisé dans le premier exercice (fichier verbs_corpus.txt)

⇒ **Pour l'exercice 2**, il s'agit d'associer un préfixe au radical russe afin de correspondre au mieux au sens du verbe français.

```
String[][] list = { {"entrer", "ходить", "в"}, {"sortir", "ходить", "вы"} };
```

Script (Main.java)

Les données se structurent sous la forme d'un tableau indexé numériquement présentant trois éléments: le verbe en français, le radical du verbe russe le préfixe russe. L'algorithme présente le verbe français et un radical russe, ensuite il demande à l'utilisateur de choisir un préfixe correct pour ce radical et vérifie la réponse.

⇒ **Pour l'exercice 3** et la fonction de traduction, l'utilisateur entre sa réponse en russe avec un verbe de mouvement conjugué. Pour cet exercice, le script va utiliser une extension vers une librairie CoreNLP pour le russe.

Script (Main.java, Sample.java, request_ruscorpora.py)

A partir d'une première entrée manuelle avec une phrase française et sa traduction en russe, le programme repère le verbe grâce à POS tag dans sa traduction russe et avec la réponse de l'utilisateur. En faisant l'analyse morphosyntaxique de la phrase russe du code source et avec la production de l'utilisateur, il cible le lemme, le temps et l'aspect. Au terme de l'analyse, on attend qu'il donne la valeur de ces trois attributs après comparaison entre le verbe du code source et validé et celui donné par l'utilisateur.

Exemple de résultats proposés par le script

```
Donnez la traduction pour: Elle est sortie de la piece
Она выходит из комнаты

the verb выходить is incorrect
right verb is выйти
The tense of verb is incorrect. Right tense is Past
The tense of your verb is Pres
The aspect of verb is incorrect. Right aspect is Perf
Right answer is :Она вышла из комнаты
```

Le script inclut de préciser si la réponse est bonne ou fausse et de donner la bonne réponse. Il va stocker les types d'erreur, faites par l'utilisateur.

Exemple à partir d'une erreur en lemme et temps pour la phrase 1 et erreur en temps et aspect pour la phrase 2. L'algorithme va chercher ce lemme avec ce temps ou cet aspect incorrect dans un grand corpus du russe (<https://ruscorpora.ru/new/>) dont il va tirer une version correcte qu'il va présenter à l'utilisateur. Le script "request_ruscorpora.py" permettra d'avoir un nombre infini d'exemples.

III- Les difficultés rencontrées

⇒ **Pour l'exercice 1** : Le fait qu'il n'y ait pas de corpus existant au préalable a présenté des difficultés dans le traitement des données. Accéder directement à un dictionnaire était compliqué car il n'y avait pas d'accès à la sémantique du verbe et puis la distinction d'un verbe envers un autre était difficile étant donnée leur ressemblance. En faire une distinction à partir de traits morphologiques n'offrait pas de solution. Nous avons dû créer notre propre corpus et nous débarrasser manuellement des mots rares. Également, nous n'avons pas réussi à trouver de dictionnaire gratuit au format texte approprié pour traduire le corpus de verbes russes, la plupart des dictionnaires disponibles représentent une image numérisée impossible à interroger. Au lieu de cela, nous avons dû utiliser des dictionnaires en ligne qui n'ont pas de traduction vérifiée. Tout d'abord, nous voulions utiliser Yandex translate qui a l'avantage de traduire du russe directement vers d'autres langues sans utiliser le modèle anglais. Malheureusement, une fois le code terminé, le site n'a pas permis de transmettre plusieurs requêtes à cause du captcha. Ensuite, nous avons essayé d'interroger DeepL dans ce cas, le serveur n'a pas répondu. Parfois, il y a des problèmes avec des demandes

simples de la partie client, donc probablement dans ce cas, la réponse a pris tellement de temps que le fichier n'a jamais été traduit. Enfin, nous avons dû importer le module Google translate et traduire notre fichier avec. Nous devons admettre que la qualité de la traduction est assez faible, malgré le fait que des mots rares ont été supprimés manuellement du corpus russe. Le problème principal est que les formes de verbes perfectifs et non perfectifs ont été traduites par le même verbe français, ce qui rend impossible l'utilisation d'un tel fichier dans l'exercice.

улетать	s'envoler
улететь	s'envoler
унести	emporter
уносить	emporter

exemple de la première sortie de Google translate

Nous avons dû changer la structure des données de tableau en dictionnaire afin de supprimer les doublons.

L'autre problème rencontré à cette étape est que le package google a été mis à jour et le script a cessé de fonctionner. La désinstallation n'a pas aidé, et comme il s'agit d'un nouveau problème, peu de réponses peuvent être trouvées. Donc, pour l'instant, nous avons un script terminé qui ne peut pas être testé car le module Google ne fonctionne pas. Heureusement, le fichier produit par ce script bien avant a été enregistré, nous pouvons donc l'importer dans un projet java lors des étapes suivantes.

Update 01/12/2020: This issue re-emerged lately, (apparently) caused once again by some changes on the Google translation API.

A solution is being discussed (again) in this [Github issue](https://github.com/ssut/py-googletrans/pull/237). Although there is not a definitive solution yet a Pull Request seem to be solving the problem:
<https://github.com/ssut/py-googletrans/pull/237>.

While we wait for it to be approved it can be installed like this:

```
$ pip uninstall googletrans
$ git clone https://github.com/alainrouillon/py-googletrans.git
$ cd ./py-googletrans
$ git checkout origin/feature/enhance-use-of-direct-api
$ python setup.py install
```

⇒**Pour l'exercice 3** : Des difficultés ont été rencontrées à plusieurs étapes de la construction de cet exercice. Les toutes premières ont été d'installer une extension CoreNLP pour le russe et d'en saisir le fonctionnement. Cette extension ne fait pas officiellement partie de CoreNLP et fournit une documentation très médiocre. Nous avons dû déduire comment certaines méthodes fonctionnent directement à partir des classes car il n'y avait pas d'exemples ou de description appropriées.

Lors des derniers tests, nous avons découvert que la librairie ne fonctionne pas correctement pour certains verbes, elle ne les reconnaît pas, bien qu'ils soient les plus basiques. Nous avons dû changer nos exercices pour éviter les verbes problématiques et NullPointerException erreur.

```
выйти Past Perf  
входить Pres Imp  
ехать null null  
Принесите null Perf  
поднести null null  
отходить Pres Imp  
Отнесите null Perf  
всходить Pres Imp  
съезжать Past Perf  
сойтись Past Perf  
объехать Past Perf  
обгонять Past Perf
```

L'autre problème important était l'appel de script python depuis java. Il a fallu beaucoup de temps pour modéliser correctement le processus d'exécution. Nous attirons également votre attention sur le chemin du fichier, si vous souhaitez le tester, vous devez le modifier.

```
Runtime rt = Runtime.getRuntime();  
process = rt.exec(String.format("/usr/local/bin/python3 /Users/Alena/Desktop/ALA0/request_roscorpora.py -v %s -n %s",
```

Finalement, nous avons fait un algorithme pas très efficace pour le parsing de la page de <https://ruscorpora.ru/new/>. Au moment où le script a été écrit, nous ne connaissions pas les librairies comme Scrapy, donc certaines fonctions du script auraient pu être écrites de manière plus simple.

IV- Une prise de recul

En général, nous avons atteint nos objectifs prévus. À la fin du projet, nous pouvons voir que certaines stratégies (comme l'utilisation de requests, argparse et os au lieu de Scrapy) étaient inefficaces et que certains modules comme Google translate ne sont pas fiables. Nous avons également noté que CoreNLP fonctionne trop lentement pour être une librairie appropriée pour les exercices interactifs.

Les points à améliorer sont de créer un script qui vérifiera la fréquence d'utilisation de chaque mot du corpus et supprimera les mots rares. D'autres exercices peuvent être proposés pour faire travailler les apprenants sur les verbes de mouvement russe. Il s'agit ici d'une première ébauche qui pourra être enrichie et améliorée par la suite. En effet, pour ajouter en complexité avec ce sujet, on peut à plus ou moins long terme envisager de proposer des exercices à partir des verbes de mouvement pour des expressions idiomatiques. Ou bien, on peut également envisager de travailler sur les formes homonymes des verbes de mouvement préverbés, homographes/homophones et homographes/non homophones avec une différence d'accent, mais tout cela dépasse le cadre de ce projet.