

Structure de la quête : *A*venture et mésaventures.

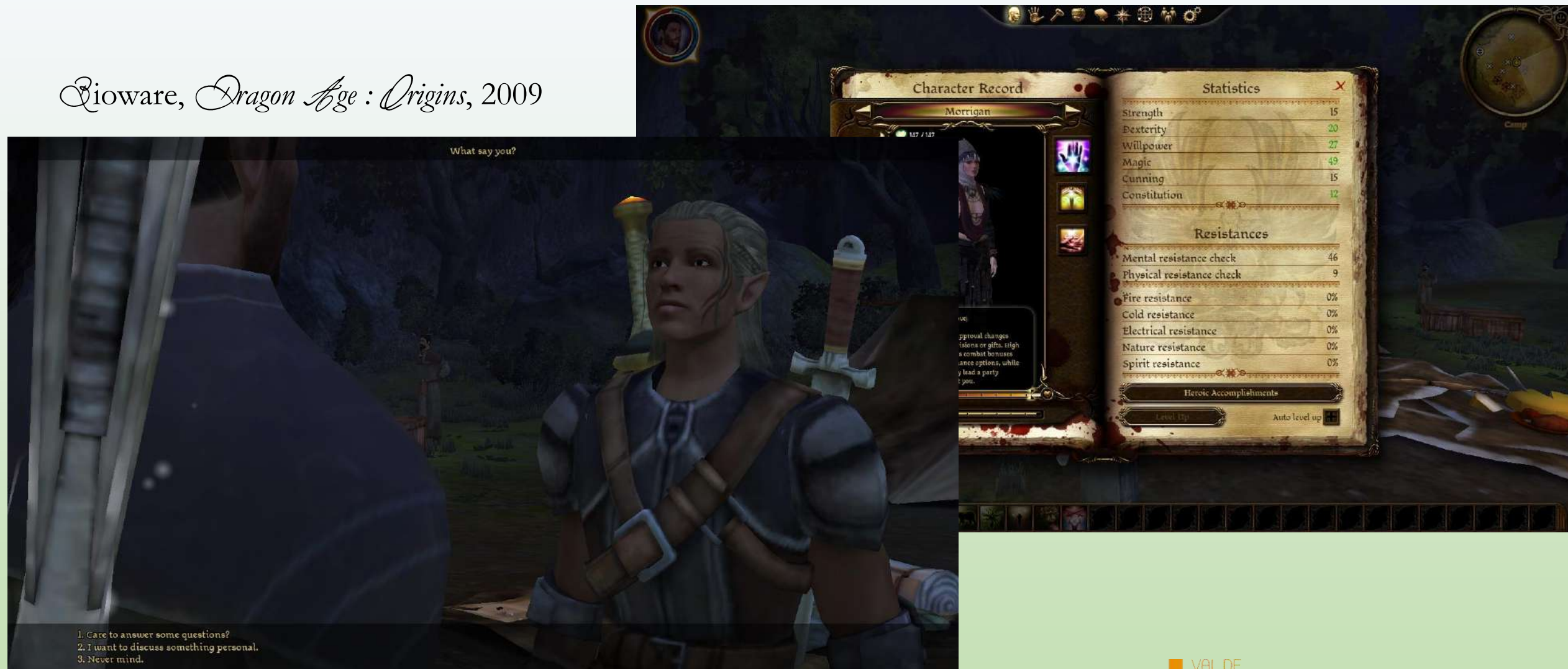
Une étude numérique des jeux vidéo *Dragon Age : Origins* et *Darkstone* à l'aune du paradigme structurel des romans d'aventures



Delphine Software, *Darkstone*, 1999

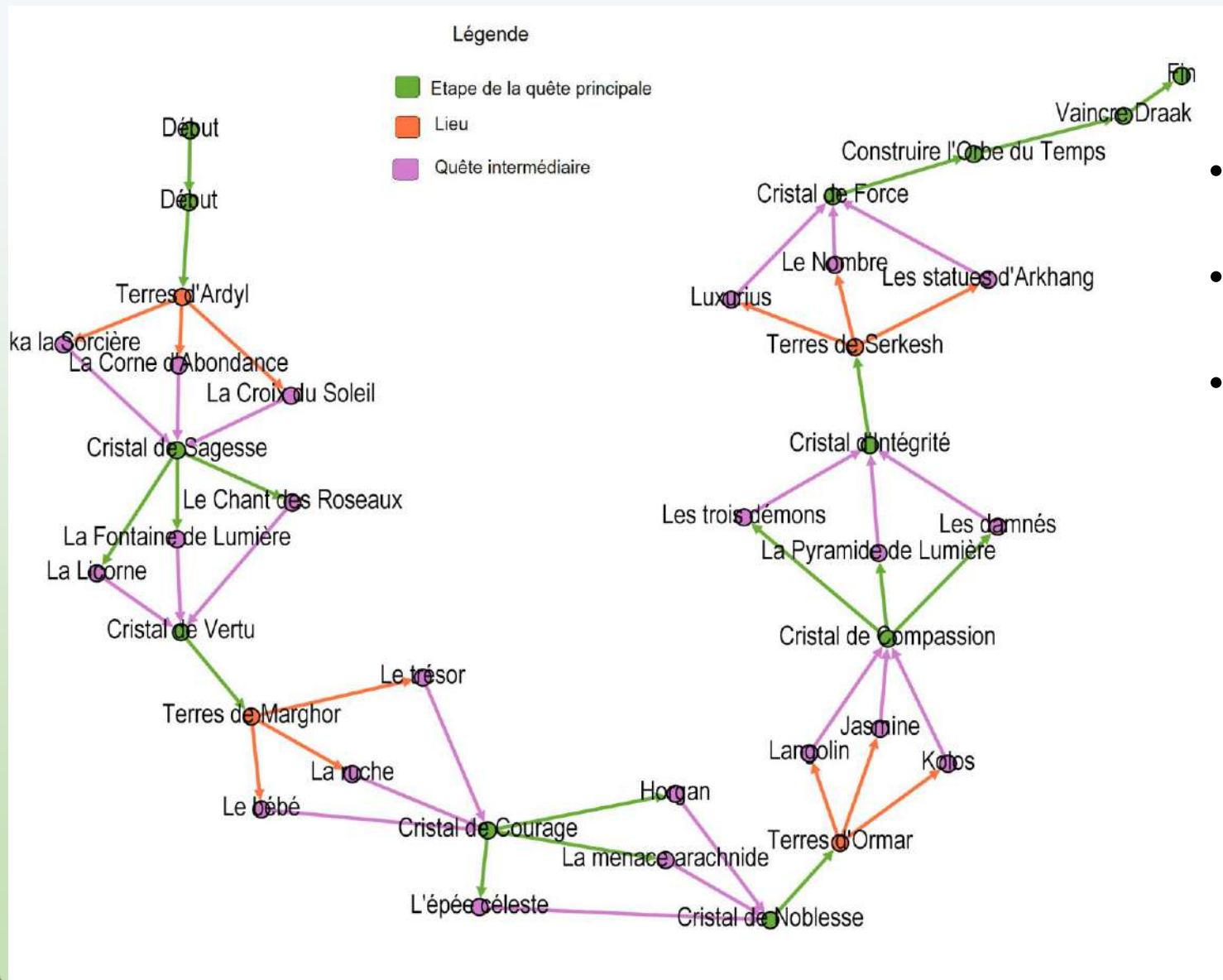


BioWare, *Dragon Age : Origins*, 2009



Paradigme structurel de l'Aventure : définir les mésaventures

Visualisation des quêtes de *Darkstone* grâce au logiciel Gephi

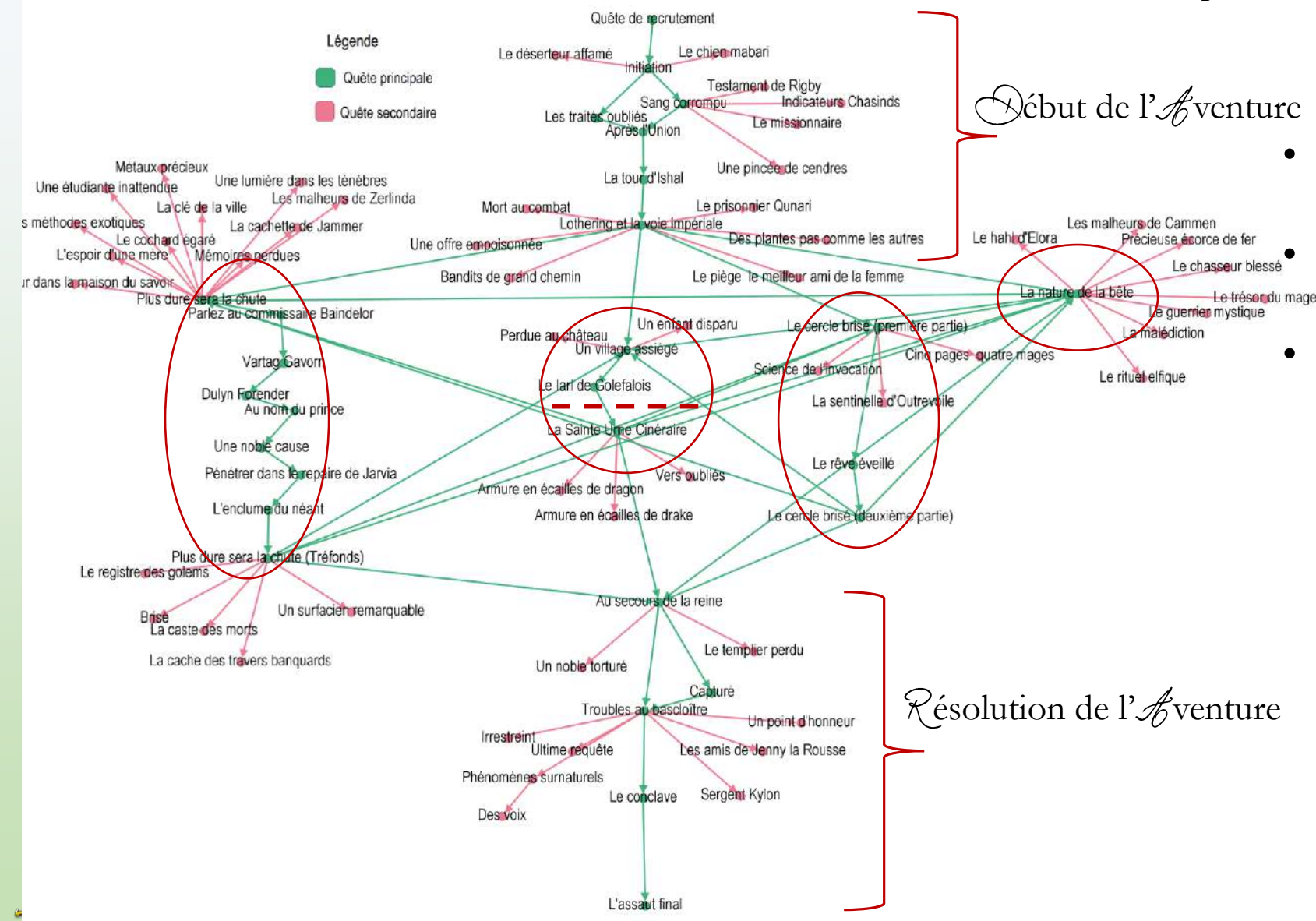


- Quête principale (Aventure) en vert
- Quêtes-jalons (mésaventures) en rose
- Quêtes facultatives : non affichées car aléatoires

Gephi (Heymann et Jacomy, 2009)

Paradigme structurel de l'Aventure : définir les mésaventures

Visualisation des quêtes de *Dragon Age : Origins* grâce au logiciel Gephi



Gephi (Heymann et Jacomy, 2009)

Paradigme structurel de l'Aventure : définir les mésaventures

Quête principale $\Leftrightarrow$ Aventure

Quête-jalons $\Leftrightarrow$ mésaventures – nécessaires
(potentiellement ordonnées dans *Darkstone*, non-ordonnées dans *Dragon Age : Origins*)

Quêtes secondaires $\Leftrightarrow$ mésaventures – facultatives (ici, non-ordonnées)

réparer les textes : pré-traitements du corpus

UTF8_EN_US.txt - Bloc-notes

```

Fichier  Edition  Format  Affichage  Aide
159      : Dragon Age
7748     : It is no trouble.
7749     : I will answer to the best of my ability.
7750     : It has been almost a year, hasn't it? Do you miss the tower at all?
7751     : You can take a mage out of the Circle, but you can't take the Circle out of the mage. Or so Irving used to
say
7752     : He was joking, but there is some truth to the statement. A mage never really leaves the Circle, and for the
rest of your life, you will be seen as a Circle mage.
7753     : Yes, even though you are a Grey Warden now, you are a mage and always will be, especially in the eyes of
others.
7754     : But you will always be a mage, especially in the eyes of others.
7755     : You represent both mages and Grey Wardens, and your actions may reflect well, or badly, on both groups.
Remember that.
7756     : That is commendable, but one's principles are often subtly influenced by others, though you may be
unconscious of it.
7757     : The only thing one can do is to be aware that this is a possibility, and ask yourself often why you do what
you do.
7758     : You do not live in isolation. Your actions have consequences and they affect those around you. Only children
and fools fail to realize this.
7759     : I am glad to hear that.
7760     : I can see that. Very well, we will no longer speak of this.
7761     : I do not want to, so try not to give me cause to do so.
7762     : Ah, but listen to me go on. You start a conversation and I just run away with it, don't I?
7763     : I've been mentoring apprentices for so long, it's become something I do unconsciously. I don't know if I
could stop even if I wanted to.
7764     : Nothing wrong with hobbling. It's a perfectly acceptable way to get around.
7765     : Oh, yes. One must be thankful for small mercies.
7766     : People don't become mages, they are born mages. The talent just surfaces later. But you are asking how I
ended up at the Circle?
7767     : I was brought there by the templars, just like many of the other apprentices. I don't remember very much. I
was very young then.
7768     : I set a boy on fire.
7769     : Oh, yes, he did. He was the eldest son of the woman who took me in. I don't think he ever liked me.
7770     : It's a long story.
7771     : I didn't have a family... I never knew my real parents. My earliest memory was of hiding in a hay loft on a
farm, trying to keep warm.
7772     : I was found, and the farmer's wife was kind enough not to send me away. But they had children of their own
and I was never made to feel welcome.

```

DP0_QUEST1.SPT - Bloc-notes

```

Fichier  Edition  Format  Affichage  Aide

QUESTNAME

{

    0 { La croix du Soleil }

    1 { The Sun Cross }

}

KEY { CROIXSOL }

QUESTINTRO

{

    0 {Le pays est vaste et il y a mille et une occasion pour de jeunes aventuriers de s'illustrer. Pour
réunir les 7 cristaux, il faudra du courage et de la persévérance. Mais celui qui y parviendra aura alors la puissance
nécessaire pour vaincre Draak et libérer le pays de son joug. Si l'aventure vous tente, vous devriez aller voir un
marchand antiquaire du nom de Darkus qui habite en dehors de la ville. On raconte qu'il propose du travail aux
aventuriers de passage. Peut-être est-ce la une occasion de vous illustrer.}

}

LAND          { 0 }

ENTRANCE      { entree }

/*****/

```

Fichiers d'origine, récupérés à partir des fichiers des jeux grâce à l'aide d'un programmeur (C++)

réparer les textes : pré-traitements du corpus

```

script_nettoyage_DAO.py
1 import re
2 import sys
3
4 with open(sys.argv[1], 'r') as f:
5     script = f.read()
6
7 #Retirer tous Les tags
8 expression = r"([A-Z]{2,})(\s|_)*([A-Z]+)*+"
9 script = re.sub(expression, r'', script, flags=re.M)
10
11 #je commence par retirer tous les numéros, ponctuations, espaces en début de ligne :
12 expression = r"[0-9]*\s*: "
13 script = re.sub(expression, r'', script, flags=re.M)
14
15 #SUPPRIMER LES _ qui traînent après suppression des majuscules.
16 expression = r"_"
17 script = re.sub(expression, r'', script, flags=re.M)
18
19 #Ensuite je sépare les répliques selon les phrases en conservant la ponctuation :
20 expression = r"(\.|\?|\!|\.\.\.)(\s)(?=[A-Z])"
21 script = re.sub(expression, r'\1\n', script, flags=re.M)
22
23 #Je supprime les espaces et les \ en trop
24 script = re.sub(r"\\", '', script, flags=re.M)
25
26 #expression5 = r'\s+(?=\s+["/\\])'
27 #script = re.sub(expression, '', script, flags=re.M)
28
29 # puis les lignes vides
30 expression = r'\n(?:\n+)'
31 script = re.sub(expression, '', script, flags=re.M)
32
33 #Sauvegarde de l'étape intermédiaire
34 with open(sys.argv[2], 'w') as fi:
35     fi.write(script)
36
  
```

- Rendre les fichiers lisibles grâce aux expressions régulières :
 - Nécessité pour *Dragon Age : Origins*
 - Impossible pour *Darkstone* : réalisé « à la main »

réparer les textes : pré-traitements du corpus

- *Darkstone*
- Tokenisation avec la librairie Nltk
- Suppression des stopwords selon la liste Nltk (fr) et suppression de la casse
- Lemmatisation avec le modèle Lefff de la librairie Spacy
- Utilisation des expressions régulières pour séparer les lignes : avoir un csv en sortie

```

script_token_lemma_csv_DS.py
1 import sys
2 import nltk
3 import re
4 import spacy
5 from spacy_lefff import LefffLemmatizer, POSTagger
6 spacy.load('fr')
7
8 nltk.download('punkt')
9 nltk.download('stopwords')
10 #ou nltk.download('popular')
11
12 with open(sys.argv[1], 'r') as f:
13     text = f.read()
14     text = re.sub(r'(?=^)', r'12345', text, flags=re.M)
15
16 tokenizer = nltk.RegexpTokenizer(r'\w+')
17 text = tokenizer.tokenize(text.lower())
18
19 stop_words = set(nltk.corpus.stopwords.words('french'))
20
21 text = [w for w in text if not w in stop_words]
22
23 text = " ".join(text)
24
25 nlp = spacy.load('fr')
26 french_lemmatizer = LefffLemmatizer()
27 nlp.add_pipe(french_lemmatizer, name='lefff')
28 text = nlp(text)
29
30 #Export des résultats
31 with open(sys.argv[2], 'a', encoding='utf-8') as fi:
32     for t in text:
33         doc = (t.lemma_)
34         fi.write(doc)
35         fi.write(" ")
36
37 with open(sys.argv[2], 'r') as fi:
38     text = fi.read()
39     text = re.sub(r'(?=12345)', '\n', text, flags=re.M)
40     with open(sys.argv[3], 'w') as fi:
41         fi.write(text)
  
```

Natural Language Processing with Python (Bird, Loper et Klein, 2009).

spaCy 2 (Honnibal et Montani, 2017)

Lefff, (Sagot, 2010).

Préparer les textes : pré-traitements du corpus

Dragon Age Origins :

- Tokenisation avec la librairie Nltk et suppression de la casse
- Suppression des stopwords selon la liste Nltk (en)
- Utilisation des expressions régulières pour séparer les lignes : avoir un csv en sortie
- Tokenisation et lemmatisation séparées à cause de la longueur du texte

```

script_token_DAO.py
1 import sys
2 import nltk
3 import re
4
5 nltk.download('punkt')
6 nltk.download('stopwords')
7 #ou nltk.download('popular')
8
9 with open(sys.argv[1], 'r') as f:
10     text = f.read()
11
12 tokenizer = nltk.RegexpTokenizer(r'\w+')
13 text = tokenizer.tokenize(text.lower())
14
15 stop_words = set(nltk.corpus.stopwords.words('english'))
16
17 text = [w for w in text if not w in stop_words]
18
19 text = " ".join(text)
20
21 text.replace('emp', '')
22 text.replace('act', '')
23 text.replace('desc', '')
24
25 ##Pour CSV
26 text = re.sub(' ', r'\n', text, flags=re.M)
27 expression = r"([0-9]+)"
28 text = re.sub(expression, r'\n1;', text, flags=re.M)
29
30
31 with open(sys.argv[2], 'w') as fi:
32     fi.write(text)
  
```

Natural Language Processing with Python (Bird, Loper et Klein, 2009).

Préparer les textes : pré-traitements du corpus

```
script_lemma_DAO.py
1 import sys
2 import os
3 from os import listdir
4 import nltk
5 from nltk.stem import WordNetLemmatizer
6 import re
7 import spacy
8
9 liste = listdir('DecoupageDAO_lemma/Decoupage_lemma_DAO_OK/')
10 liste.sort()
11 print (liste)
12
13 for l in liste :
14     with open('DecoupageDAO_lemma/Decoupage_lemma_DAO_OK/'+l, 'r') as l:
15         #Input un csv déjà tokenizé et sans les stopwords = x4 à cause de la longueur du texte
16         text = l.read()
17
18         nlp = spacy.load('en')
19         resultat = nlp(text)
20
21         with open(sys.argv[1], 'a') as fi:
22             for r in resultat:
23                 #doc = (t.text, t.pos, t._.melt_tagger, t._.lefff_lemma, t.tag_, t.lemma_)
24                 doc = (r.lemma_)
25                 fi.write(doc)
26                 fi.write(" ")
27 #csv en output
28
```

- *Dragon Age Origins* :
- Pré-découpage du fichier à cause de sa longueur
- Boucle pour le traitement de l'ensemble des parties
- Lemmatisation avec le modèle anglais de la librairie Spacy

spaCy 2 (Honnibal et Montani, 2017)

réparer les textes : pré-traitements du corpus

Textes tokenisés, lemmatisés et sans stopwords

1	trouble
2	answer good ability
3	almost year
4	miss tower
5	take mage circle take circle mage
6	irving use say
7	joke truth statement
8	mage never really leave circle rest life see circle mage
9	yes even though grey warden mage always especially eye other
10	always mage especially eye other
11	represent mage grey warden action may reflect well badly group
12	remember
13	commendable one principle often subtly influence other though may unconscious
14	thing one aware possibility ask often
15	live isolation
16	action consequence affect around
17	child fool fail realize
18	glad hear
19	see
20	well longer speak
21	want try give cause
22	ah listen go
23	start conversation run away
24	mentor apprentice long become something unconsciously
25	know could stop even want
26	nothing wrong hobbling
27	perfectly acceptable way get around
28	oh yes
29	one must thankful small mercy
30	people become mage bear mage
31	talent surface later
32	ask end circle
33	bring templar like many apprentice
34	remember much
35	young
36	set boy fire
37	oh yes
38	eld son woman take
39	think ever like
40	long story
41	family
42	never know real parent
43	early memory hide hay loft farm try keep warm
44	find farmer wife kind enough send away
45	child never make feel welcome

A	B
1	1bel exploit
2	2vous trouver cristal sagesse
3	3mais rester encore six autre cristal découvrir afin prouver bien élu dont parler prophétie
4	4quand récupérer tout cristal apporter
5	5je pouvoir alors invoquer puissance kaliba faire apparaître orbe temps
6	6a bientôt bon chance
7	7maître draak sentir très agacer misérable petit croisade
8	8il avoir charger éliminer récupérer trois cristal
9	9ne résister si souhaiter mort douce rapide
10	10ah ah ah
11	11pauvre humain
12	12ce cristal aucun utilité
13	13ce vieux fou sébastien surestimer bien trop pouvoir
14	14invincible bientôt dieu
15	15et place monde préparer
16	16le pays vaste avoir mille occasion jeune aventurier illustrer
17	17pour réunir sept cristal falloir courage persévérance
18	18mais celui parviendra alors puissance nécessaire vaincre draak libérer pays joug
19	19si aventure tenter devoir aller voir marchand antiquaire nom drakus habiter dehors ville
20	20on raconter proposer travail aventurier passage
21	21peut être là occasion illustrer
22	22il exister issue secret sanctuaire kaliba mener tombeau enoch
23	23paix enfant kaliba
24	24le seigneur draak venger
25	25bonjour chez
26	26bonjour chez
27	27drakus aigrefin
28	28si jour affaire méfier
29	29on raconter impliquer bon nombre histoire louche
30	30que châtié criminel avoir oser voler pomme arbre nourrir famille
31	31découvrir clé regretter incompétence
32	32fermer clé
33	33maintenant achever criminel porte ouvrir
34	34fermer clef
35	35verrouiller
36	36bien
37	37toujours vie donc faire bon choix
38	38pour devenir disciple déesse kaliba falloir réussir deux épreuve
39	39franchir ce porte affronter destin
40	40tu vaincre épreuve
41	41maintenant choisir arme revenir voir
42	42comme bizarre aussi choisir ce épée bois sans aucun valeur
43	43te voilà disciple kaliba présent
44	44porte ce amulette déesse reconnaître
45	45va naix enfant kaliha

Réaliser une étude de sujet : le *topic-modeling*

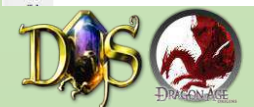
```

1-|---
2 title: "Mallet_with_R"
3 author: "Jean-Baptiste Camps, modifié par Alice Dionnet"
4 date: "3-9 février 2020 modification Avril 2020"
5 output: html_document
6 ---
7 install.packages("cowplot")
8 install.packages("dplyr")
9 install.packages("factoextra")
10 install.packages("ggpubr")
11 install.packages("rJava")
12 library("rJava")
13 library("mallet")
14 library("knitr")
15 library("factoextra")
16 install.packages("deducr",, 'http://www.rforge.net/', type='source')
17 library("deducr")
18
19 ---[r setup, include=FALSE]
20 knitr::opts_chunk$set(echo = TRUE)
21 ---
22 ## création des modèles de sujet
23
24
25 ---[r]
26 d = read.csv(file = "DAO_lemma.tsv", sep = "\t", quote = '\'', header = TRUE, encoding="UTF-8")
27 # On a maintenant
28 nrow(d) #échantillons
29 # import des documents
30 library("mallet")
31 docs = mallet::mallet.import(rownames(d), as.character(d[,1]), "DAO_stopwords.txt", token.regex = "[\\s]+")
32
33
34 # Créons le modèle à entraîner:
35
36 ---[r]
37 topic.model <- MalletLDA(num.topics = 5, alpha.sum = 1, beta = 0.1)
38
39 #Charger un modèle
40
41 ---[r]
42 topic.model <- load(file = "modeleDAO_topics_quete_environnement_spirituel_queteprincipale_actions")
43
44 #chargeons les docs
45 ---[r]
46 topic.model$loadDocuments(docs)
47
48
49 # Optimisation des hyperparamètres (\code{alpha} and \code{beta}) toutes les 20 itérations, après 50 itérations initiales.
50
51 ---[r]
52 topic.model$setAlphaoptimization(20, 50)
53
54
55 # Now train a model. Note that hyperparameter optimization is on, by default. We can specify the number of iterations. Here we'll use a large-ish round number.
56
57 ---[r]
58 topic.model$train(500)
59 #topic.model$train(100)
60 # On peut regarder comment les valeurs ont changé
61 # topic.model$alpha()
62 # topic.model$beta()
63
64
65
66 ## Visualisation interactive avec LDAvis
67
68 ---[r]
69 #install.packages("LDAvis")
70 # si besoin
71 #install.packages("servr")
72 library("servr")
73 library("LDAvis")
74 source("LDAvis_function.R")
75 makeLDAvis(topic.model, outDir = tempFile(), openBrowser = TRUE)
76
77
78 ## Sauvegarde d'un modèle
79
80 ---[r]
81 save(topic.model, file = "modeleDAO")
82
83
  
```

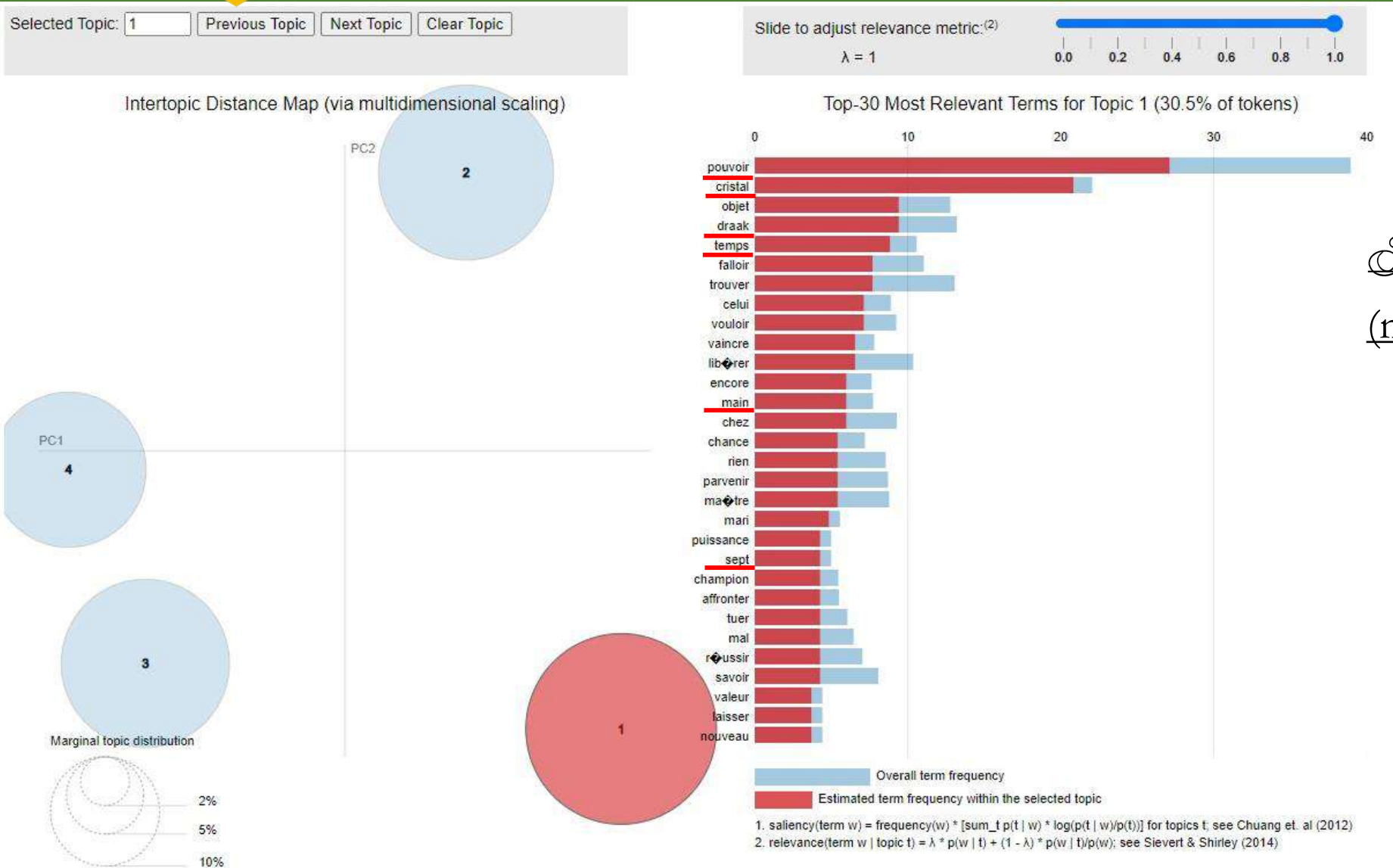
- *Dragon Age Origins* :
- Utilisation de la librairie Mallet
- Variation du nombre de sujets recherchés : résultats apparemment les plus pertinents avec cinq sujets
- Ajout de *stopwords* selon les résultats obtenus, au fil des tests
- Visualisation grâce à la librairie LDAvis
- *Darkstone* :
- Même procédure, résultats plus pertinents avec quatre sujets

LDAvis (Sievert et Shirley, 2014)

MALLET (McCallum, 2002)



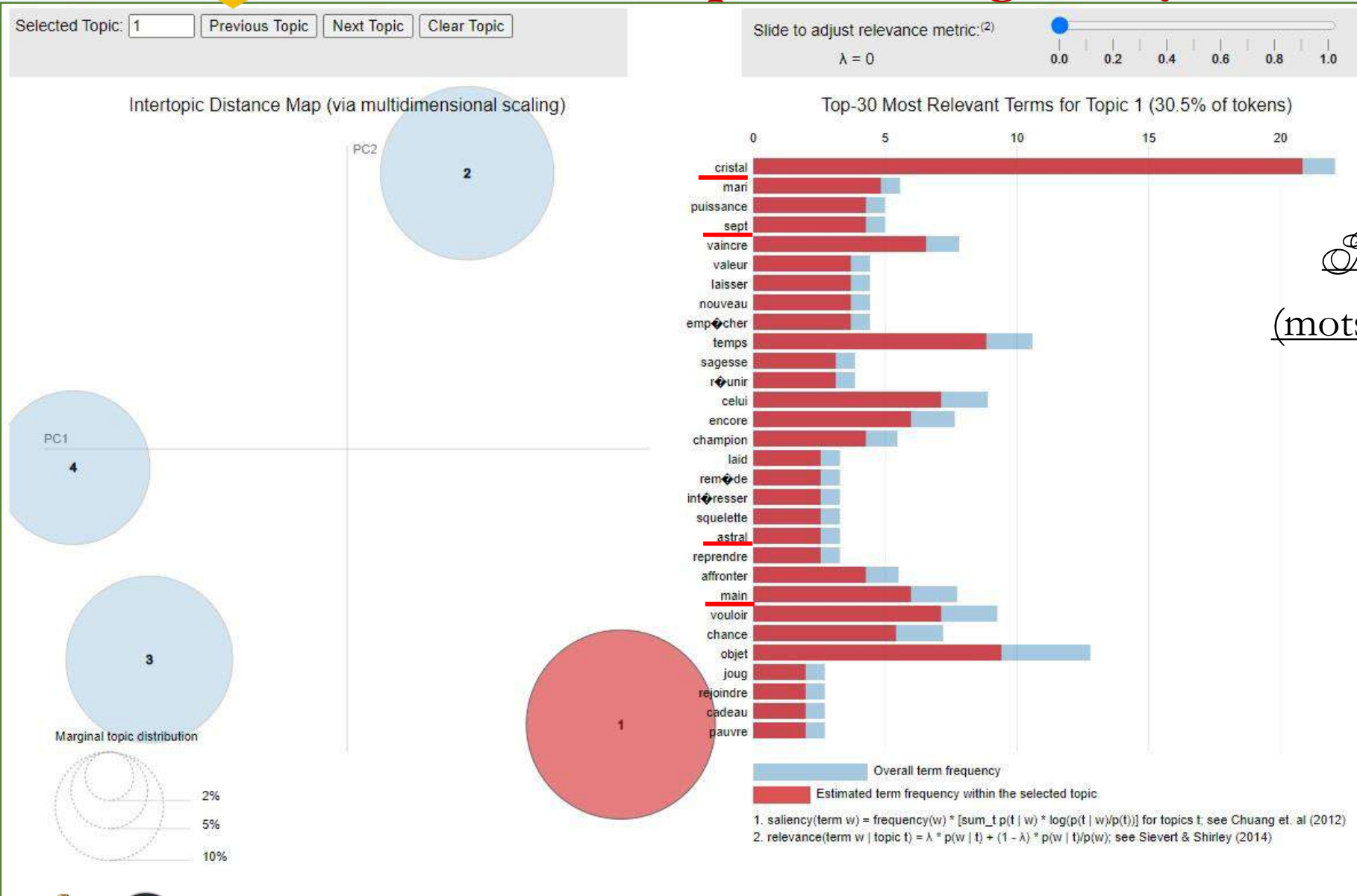
Topic modeling : analyse des résultats



Premier sujet de Darkstone
(mots les plus fréquents, $\lambda = 1$)

- Pouvoir
 - Cristal
 - Draak
 - Temps (Orbe du temps ?)
 - Main (Main Astrale ?)
 - Sept (Sept cristaux ?)
- = Quête principale

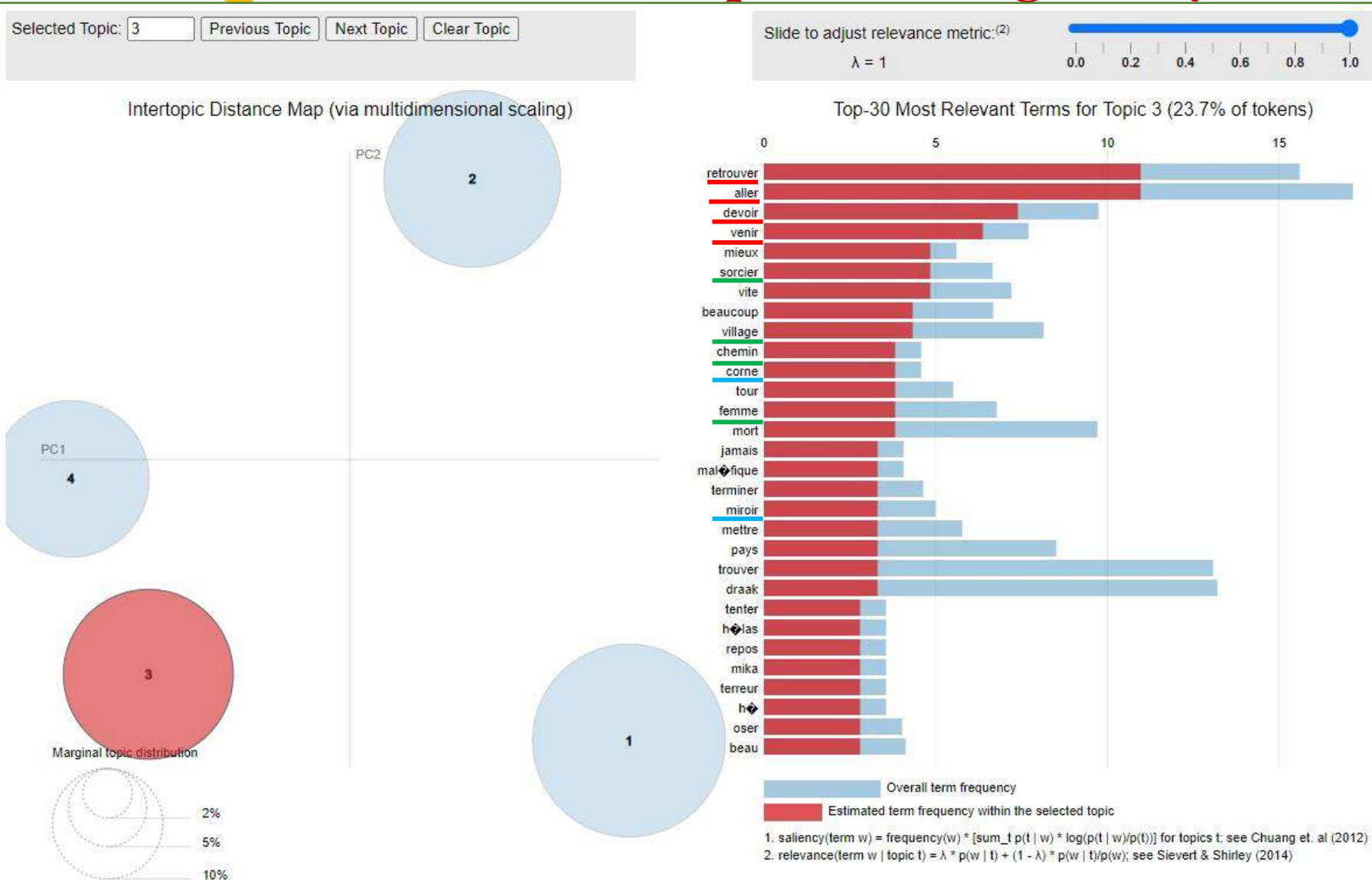
Topic modeling : analyse des résultats



Premier sujet de Darkstone
(mots les plus caractéristiques, $\lambda = 0$)

- Cristal
 - Sept
 - Astral
 - Main
- = Quête principale

Topic modeling : analyse des résultats



Troisième sujet de Darkstone
(mots les plus fréquents, $\lambda = 1$)

- Retrouver
- Aller
- Devoir
- Venir

= Termes de quêtes génériques

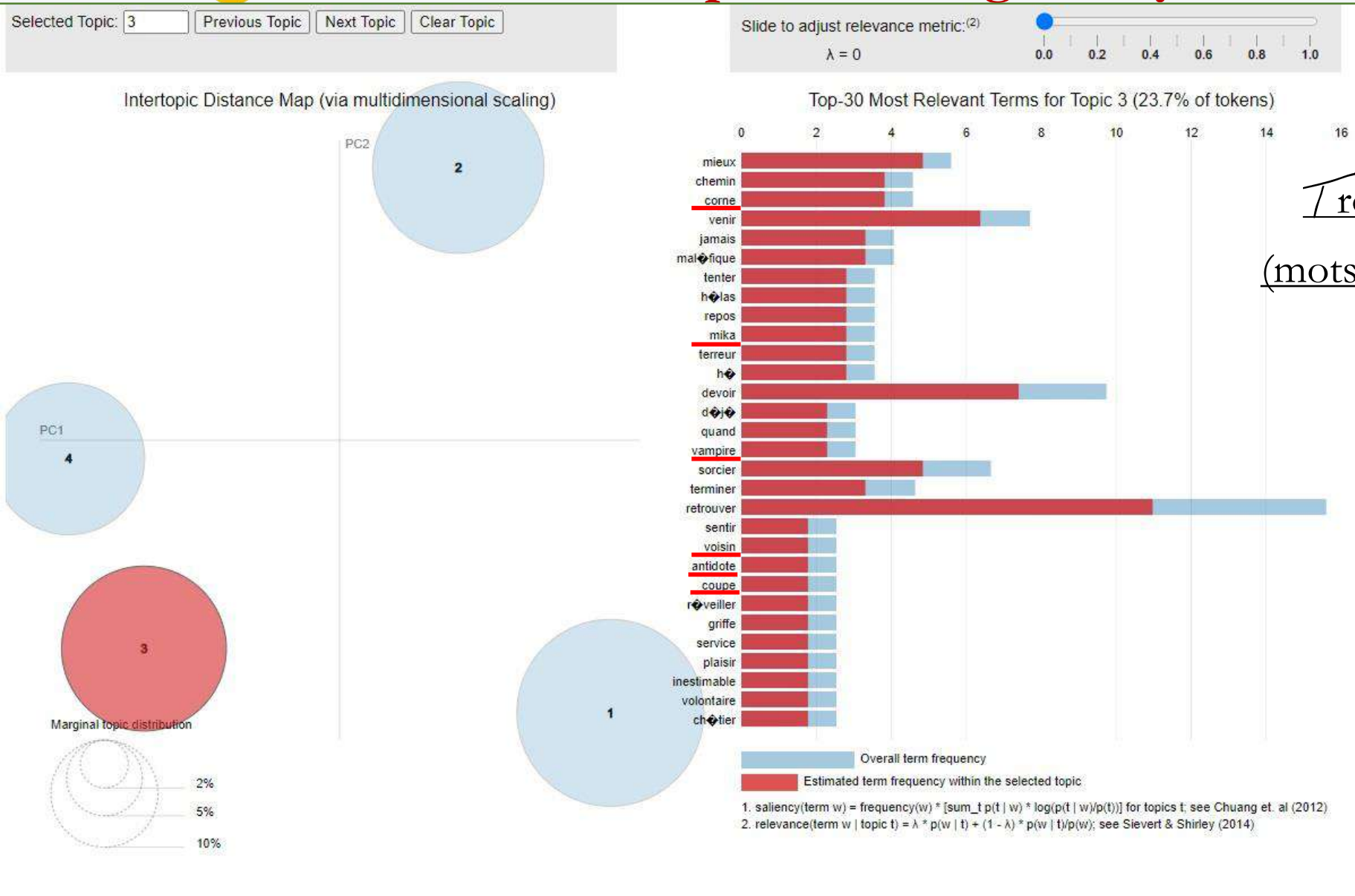
- Sorcier
- Village
- Femme

= Présents dans de nombreuses quêtes-jalons

- Corne
- Miroir

= Spécifiques à deux quêtes-jalons

Topic modeling : analyse des résultats



Troisième sujet de Darkstone
(mots les plus caractéristiques, $\lambda = 0$)

- Corne
 - Miroir
 - Vampire
 - Voisin
 - Antidote
 - Coupe
- = Représentent cinq quêtes-jalons

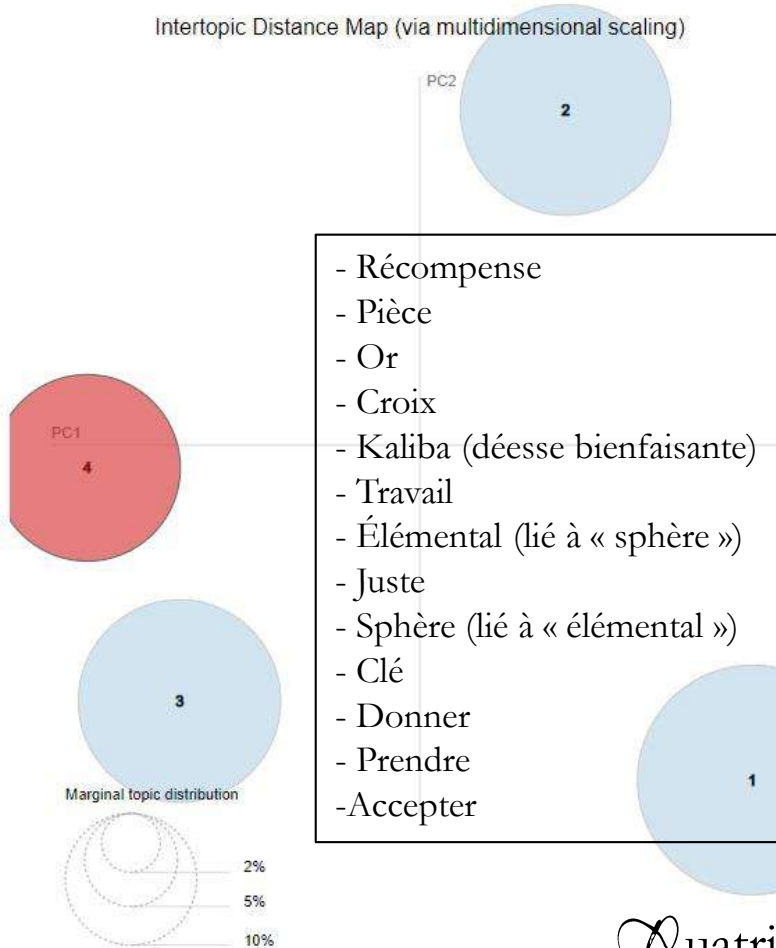
Topic modeling : analyse des résultats

Selected Topic: Previous Topic Next Topic Clear Topic

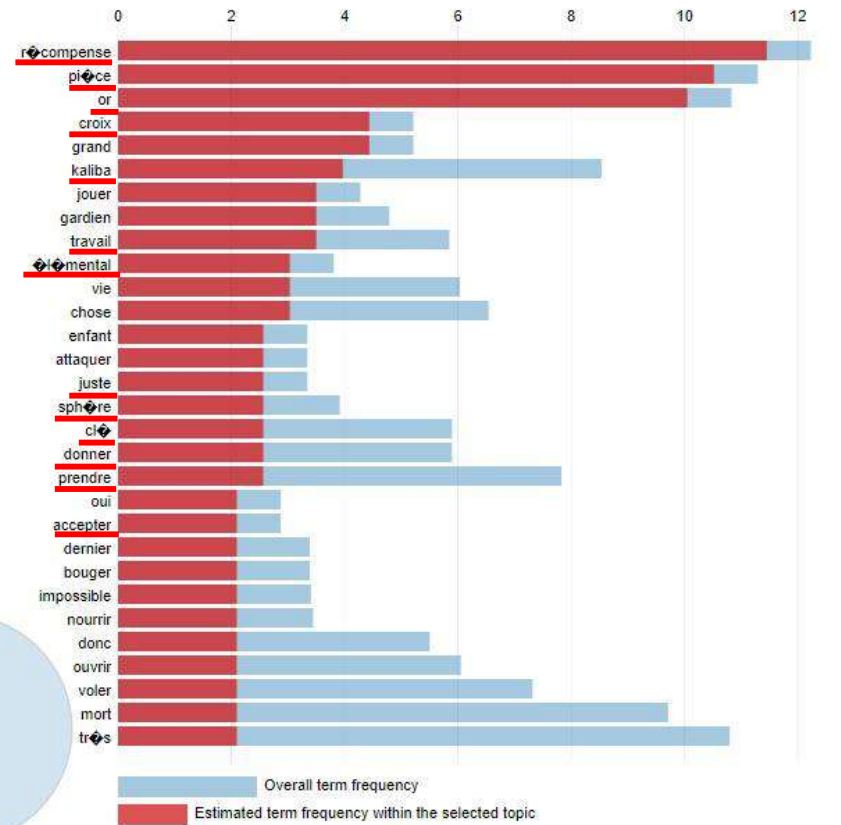
Slide to adjust relevance metric:⁽²⁾
 $\lambda = 1$ 0.0 0.2 0.4 0.6 0.8 1.0

Slide to adjust relevance metric:⁽²⁾
 $\lambda = 0$ 0.0 0.2 0.4 0.6 0.8 1.0

Intertopic Distance Map (via multidimensional scaling)

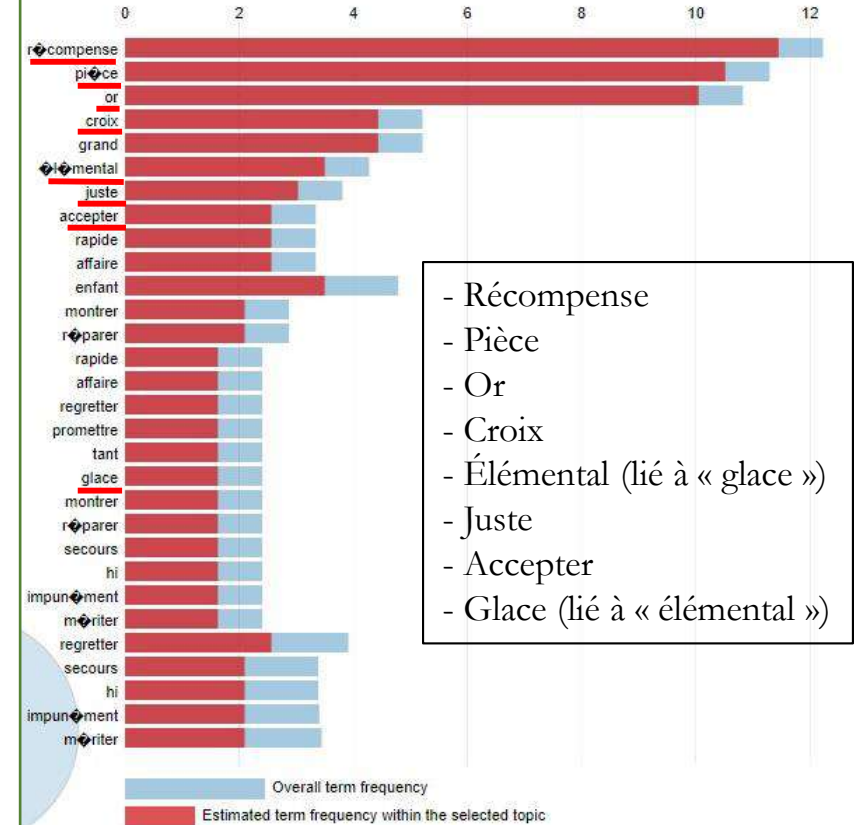


Top-30 Most Relevant Terms for Topic 4 (20.1% of tokens)



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))]; for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = $\lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

Top-30 Most Relevant Terms for Topic 4 (20.1% of tokens)



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))]; for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = $\lambda * p(w | t) + (1 - \lambda) * p(w | t)/p(w)$; see Sievert & Shirley (2014)

- Récompense
- Pièce
- Or
- Croix
- Élémental (lié à « glace »)
- Juste
- Accepter
- Glace (lié à « élémental »)

Quatrième sujet de Darkstone

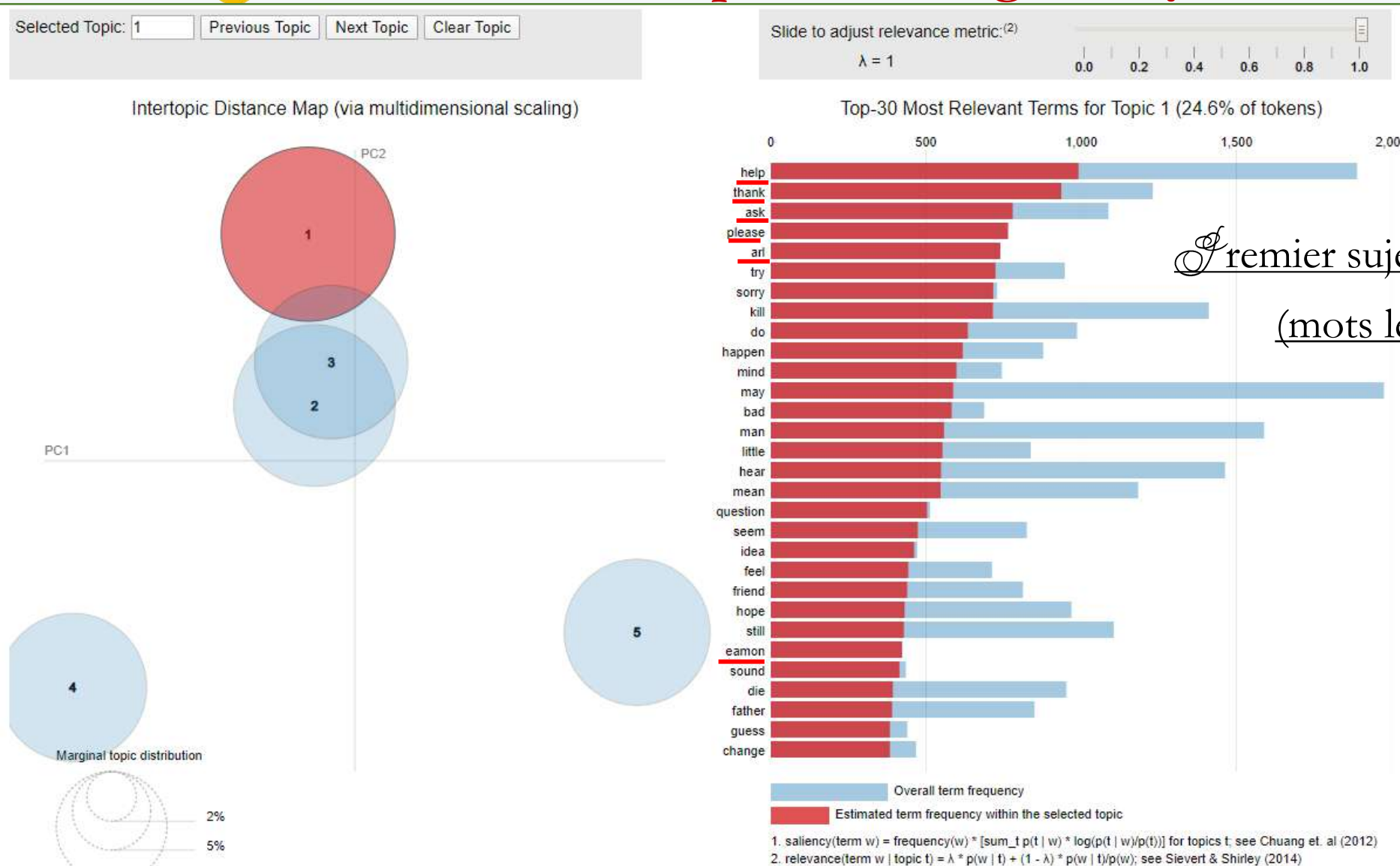
(mots les plus fréquents, $\lambda = 1$ à gauche, mots les plus caractéristiques, $\lambda = 0$ à droite)



Colloque « Les données de la recherche au temps des humanités numériques »



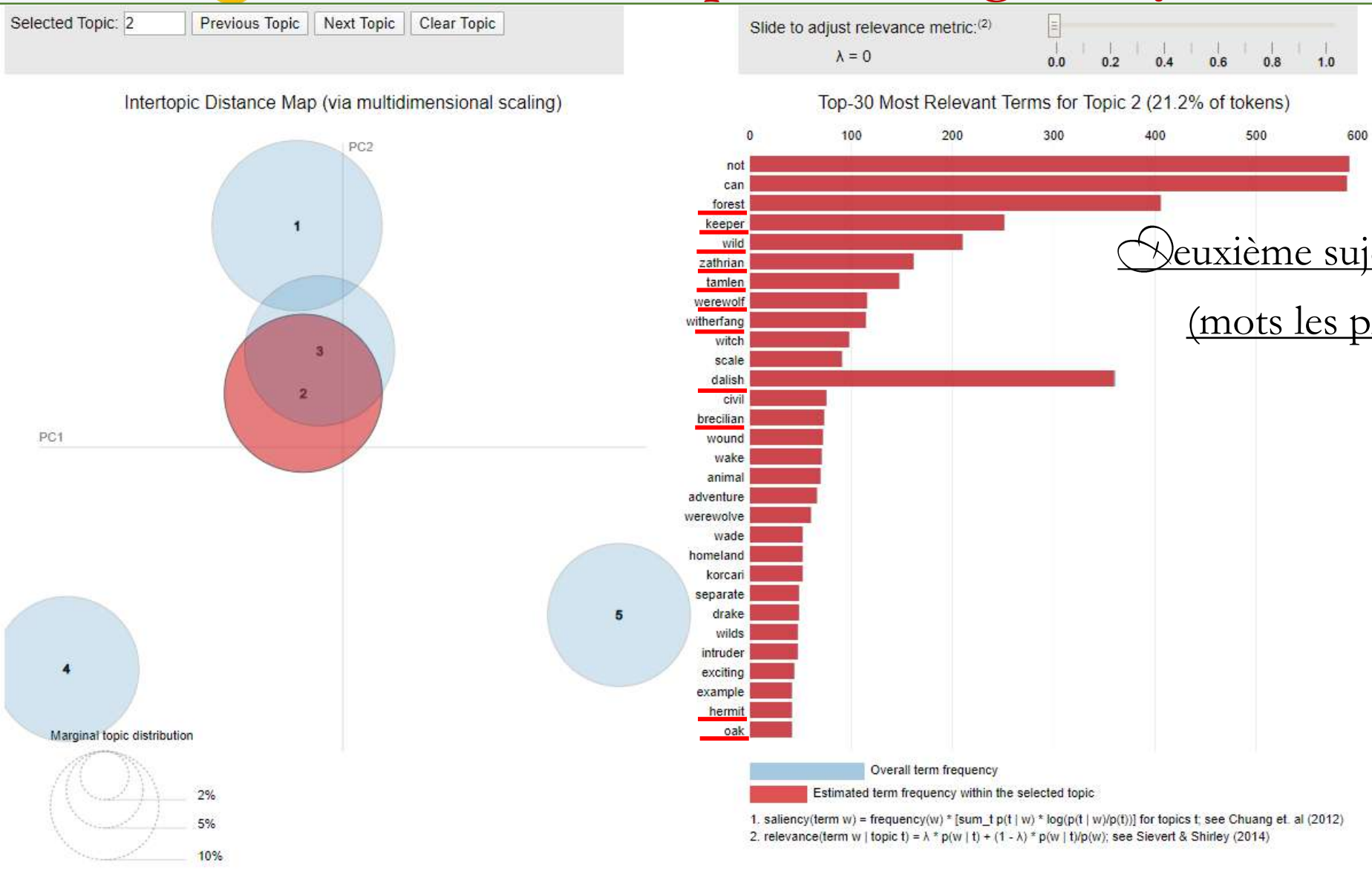
Topic modeling : analyse des résultats



Premier sujet de Dragon Age : Origins
 (mots les plus fréquents, $\lambda = 1$)

- Help
 - Thank
 - Ask
 - Please
 - Arl
 - Eamon
- = Quête-jalon « 1 »

Topic modeling : analyse des résultats

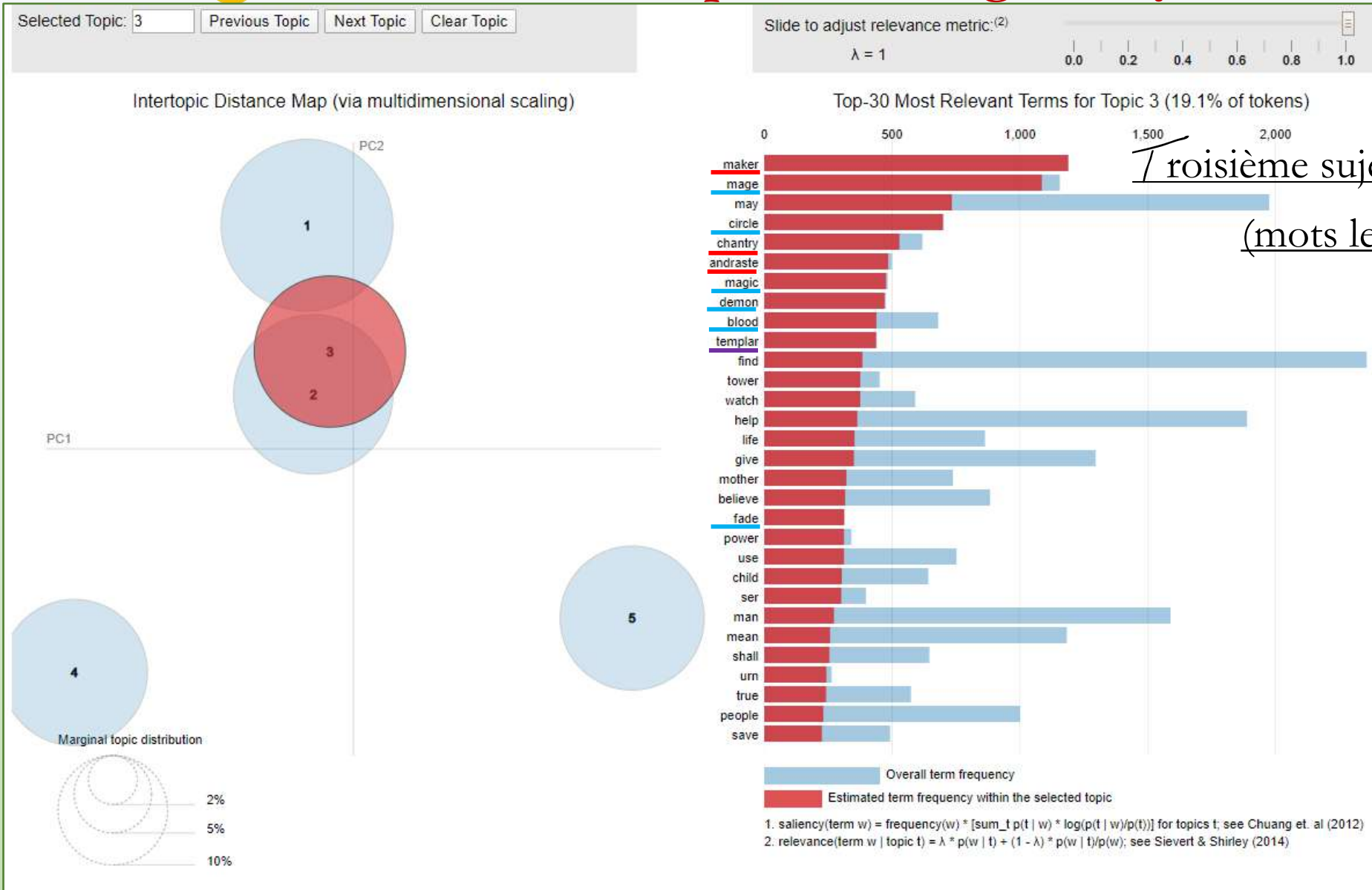


Deuxième sujet de Dragon Age : Origins

(mots les plus caractéristiques, $\lambda = 0$)

- Forest
 - Keeper
 - Wild
 - Zathrian
 - Tamlen
 - Werewolf
 - Witherfang
 - Dalish
 - Brecilian
- = Quête-jalon « 2 »

Topic modeling : analyse des résultats



Troisième sujet de Dragon Age : Origins

(mots les plus fréquents, $\lambda = 1$)

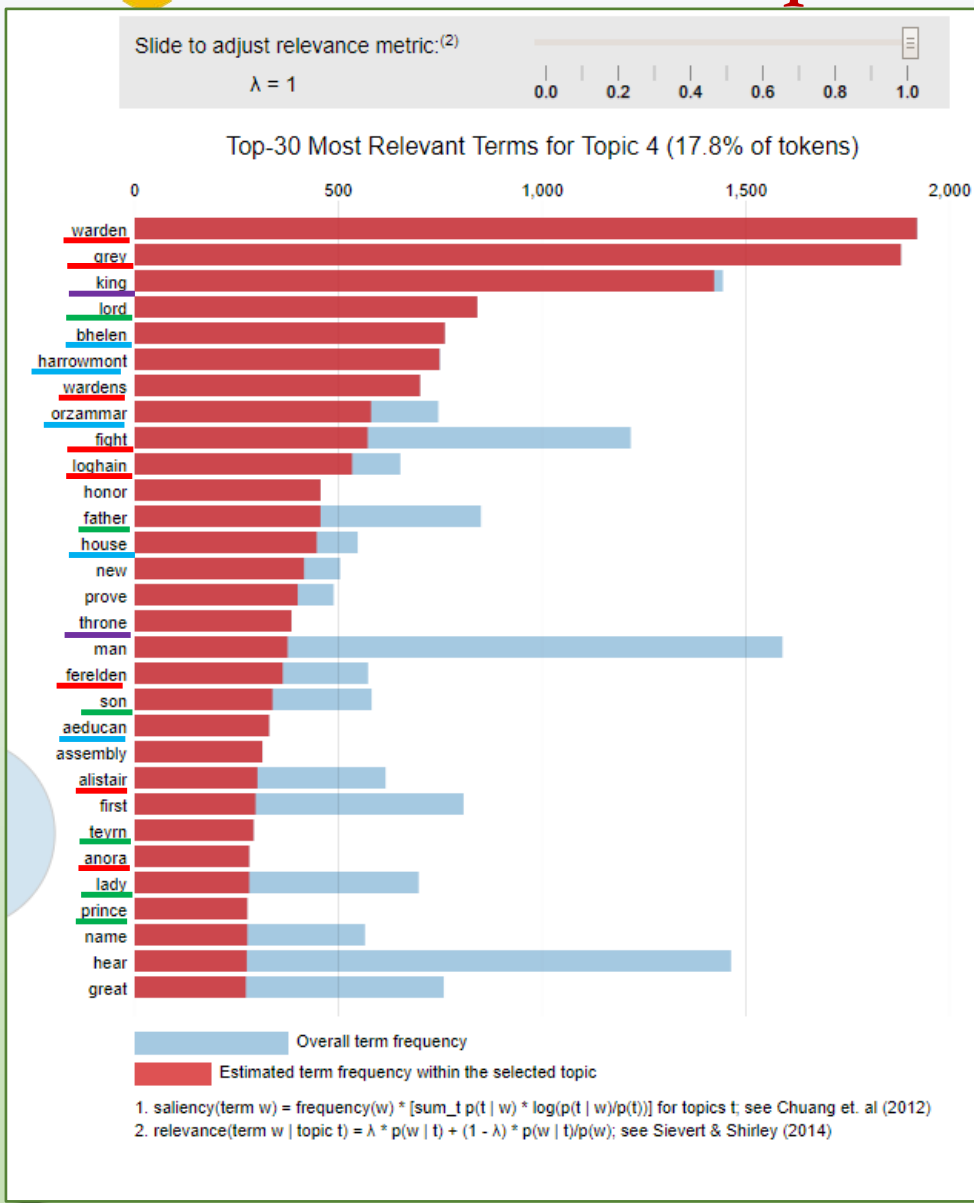
- Maker
 - Chantry
 - Andraste
 - Templar
- = Quête-jalon « 1,5 »

- Mage
 - Circle
 - Magic
 - Demon
 - Blood
 - Fade
- = Quête-jalon « 3 »

Topic modeling : analyse des résultats

Quatrième sujet de Dragon Age : Origins

(mots les plus fréquents, $\lambda = 1$)



- Bhelen
 - Harrowmont
 - Orzammar
 - House
 - Aeducan
 - King
 - Throne
- = Quête-jalon « 4 »

- Warden
 - Grey
 - Fight
 - Loghain
 - Ferelden
 - Alistair
 - Anora
 - King
 - Throne
- = Quête principale

- King
 - Lord
 - Father
 - Throne
 - Son
 - Teyrn
 - Lady
 - Prince
- = Politique

Conclusion

Résultats d'ensemble

 Quêtes-jalons aléatoires dans *Darkstone* : renforce l'aspect interchangeable et la déconnexion par rapport à l'*Aventure*.

- Topic modeling => premier sujet « quête principale » et troisième sujet « quêtes-jalons » dans *Darkstone* : séparation des deux et proximité des sujets 3 et 4 (« récompense »).

 *Dragon Age : Origins* => pas d'ordre dans les quêtes-jalons, qui font progresser l'histoire par leur dénouement ; quêtes secondaires déconnectées de l'*Aventure*.

- Topic modeling => *Dragon Age : Origins* distingue « quêtes guerrières » (premier, deuxième et troisième sujets) et « quêtes politiques » (quatrième sujet).

= Une séparation plus thématique dans *Dragon Age : Origins* et plus structurelle dans *Darkstone*

= Respect de la structure en épisodes typique des romans d'aventures visible tant dans la structure narrative que via l'analyse des sujets

Conclusion

Thèse

- Proposer une étude de la narration interactive qui soit indépendante du support et diachronique.
- Proposer différents genres de la narration interactive, de la même manière qu'il existe des genres narratifs ou ludiques.
- Restreindre cette étude autour des modèles narratifs de « quête » et « enquête », qui sont particulièrement propices à la narration interactive
- Proposer cette typologie à partir d'un corpus d'une vingtaine d'œuvres, comparées notamment grâce à la classification non-supervisée.