PLDAC 2015 / Sujet 6
Optimisation de l'extraction de résumés linguistiques flous dans des données numériques

G.Moyse

20 mars 2015

1 Rappel du sujet

Les résumés linguistiques flous [1, 2] permettent la génération de phrases descriptives d'ensembles de données numériques. Ils présentent un certain nombre d'avantages par rapport aux approches statistiques, en termes d'interprétabilité notamment. Une question ouverte aujourd'hui concernant ces résumés est la mise au point d'une méthode efficace de parcours de l'ensemble des résumés possibles pour un jeu de données et un vocabulaire donnés. Un certain nombre de résultats théoriques permettant d'optimiser ce parcours ont été proposés par l'équipe. L'objectif de ce stage est d'en réaliser l'implémentation et d'en analyser les résultats.

2 Contexte

Les éléments liés à la logique floue et aux résumés linguistiques flous nécessaires à la bonne exécution du projet sont présentés dans les deux sous-parties suivantes.

2.1 Éléments de logique floue

Une introduction complète à la logique floue est disponible ici [3]. D'autre part, l'UE MORACOI [1] dispensé dans le master DAC aborde également ces questions. Dans le cadre de ce projet, seules les notions décrites dans les paragraphes suivants seront nécessaires.

2.1.1 Sous-ensemble flou

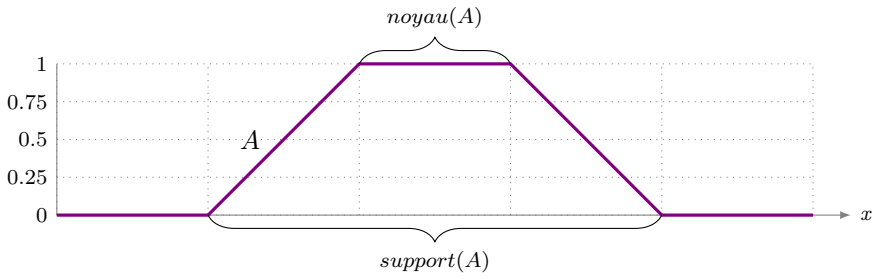

FIGURE 1 – Le sef A, son noyau, son support

Contrairement à la théorie classique des ensembles où un élément d'un univers appartient ou pas à un ensemble, en logique floue, un élément peut appartenir un peu, complètement, ou pas du tout à un *sous-ensemble flou* (sef) [4].

On note l'appartenance d'un élément à un sef comme un nombre dans $[0, 1]$. Par exemple, un élément peut appartenir à 0,4 à un sef.

Formellement, un sef A possède une *fonction d'appartenance* $A(x)$ définie dans [0,1]. Dire que x appartient à A à un degré 0,4 s'écrit $A(x) = 0, 4$.

1. http ://dac.lip6.fr/master/enseignement/ues/moracoi/

Noyau et support On appelle noyau d'un sef A l'ensemble des points qui y appartiennent complètement. Formellement, $noyau(A) = \{x \in X | A(x) = 1\}$.

On appelle support d'un sef A l'ensemble des points qui y appartiennent au moins un peu. Formellement, $support(A) = \{x \in X | A(x) > 0\}$.

Ces différentes notions sont illustrées sur la Fig. 1

2.1.2 Variable linguistique

Les sef peuvent être utilisés pour définir des *modalités* de *variables linguistiques* (VL) [5]. La Fig. 2 illustre la VL *Taille* avec ses 3 modalités *Petit*, *Moyen* et *Grand*. Une taille donnée appartient plus ou moins à l'une de ces trois modalités. Par exemple, 95 cm appartient à 0,5 à *Petit*, 0,5 à *Moyen* et 0 à *Grand*. 180 cm appartient à 0 à *Petit*, 0 à *Tiède* et 0 à *Chaud*. Une taille $<$ 60 cm appartient à *Petit* à 1, et $>$ à 210 cm à *Grand* à 1.

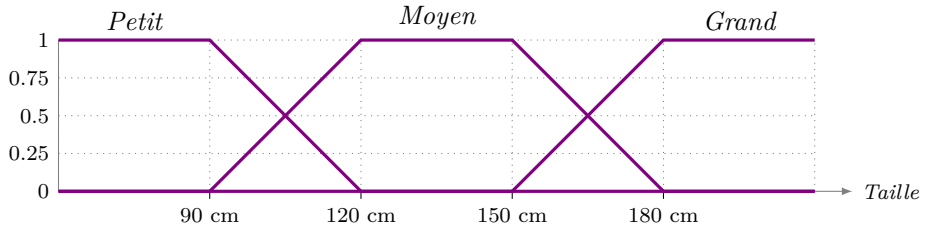

FIGURE 2 – La VL "Taille"

VL sur un univers discret La VL *Taille* est définie sur l'univers continu des tailles, considérées ici entre 60 et 210 cm. Une VL peut également être définie sur un univers discret, comme par exemple la VL *Sexe* contenant les modalités *Homme* et *Femme* sur l'univers $\{H, F\}$, ou la VL *Saison* avec les modalités *Printemps*, *Ete*, *Automne*, *Hiver* sur l'univers $\{Jan, Fev, Mar, Avr, Mai, Jun, Jui, Aou, Sep, Oct, Nov, Dec\}$.

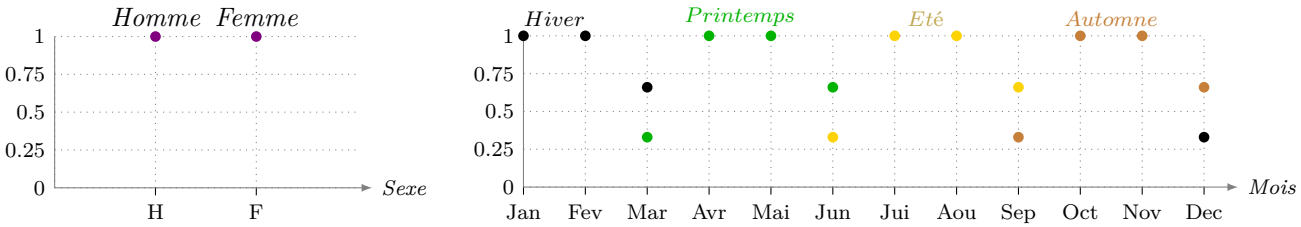

FIGURE 3 – Les VL "Sexe" et "Saisons"

2.1.3 Opérations ensemblistes en logique floue

Les opérations classiques d'*intersection* et de *cardinalité* sont aussi disponibles en logique floue.

Intersection L'intersection en logique floue est représentée par une *t-norme* notée $\top$. Pour deux éléments définis sur $\mathbb{R}$, elle est définie de $\mathbb{R}^2$ dans $[0, 1]$. Plusieurs t-normes existent, les 3 principales sont :

$$\text{t-norme de Zadeh} : \top_Z(a, b) = \min(a, b)$$
$$\text{t-norme probabiliste} : \top_P(a, b) = ab$$
$$\text{t-norme de Łukasiewicz} : \top_L(a, b) = \min(a + b - 1, 1)$$

Les intersections calculées avec la t-norme de Zadeh de *Petit* / *Moyen* et *Moyen*/ *Grand* sont représentées sur la Fig. 4.

Cardinalité La cardinalité d'un sef peut être calculée de différentes manières. Dans le cadre de ce projet, nous utiliserons la cardinalité floue classique, définie pour un sef A comme :

$$|A| = \sum_x A(x)$$

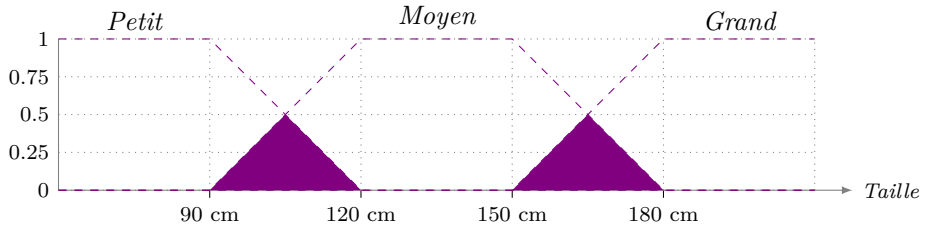

FIGURE 4 – Intersections floues calculées avec $\top_Z$ des modalités de la VL "Taille"

2.2 Les résumés linguistiques flous

2.2.1 Concepts généraux

Résumé linguistique flou Un *résumé linguistique flou* (RLF) est constitué d'un ensemble de *phrases* instanciées à partir de *protoformes*. Il décrit un ensemble de données de manière textuelle.

Phrase Une phrase d'un RLF est instanciée selon un protoforme et associée à une *valeur de vérité*. Dans les paragraphe suivants, nous utiliserons l'exemple de phrase "La plupart des jeunes sont grands" (0.88). La valeur entre parenthèses est la *valeur de vérité* et sert à mesurer à quelle point la phrase est vraie par rapport aux données étudiées.

Protoforme Un protoforme est un modèle de phrase. Les protoformes pris en compte dans le projet sont "QRx sont P", où Q représente un *quantificateur*, R un *univers* et P un *attribut*. Q, R et P sont représentés par des sef.

Dans l'exemple, le quantificateur Q est $LaPlupart$, l'univers R est $Jeunes$ et l'attribut P est $Grand$.

Quantificateur Le quantificateur mesure l'adéquation du décompte réalisé pour évaluer la phrase avec le quantificateur retenu. Dans l'exemple, le nombre de jeunes qui sont grands est comparé à "La plupart".

Les quantificateurs sont classés en *quantificateurs absolus* et *quantificateurs relatifs*.

Les quantificateurs relatifs servent à évaluer des mesures rapportées au nombre d'individus dans l'univers considéré, donc dans $[0, 1]$. Ces quantificateurs sont par exemple Peu, $EnvironLaMoitié$, $LaPlupart$ etc. (cf. Fig. 5).

Les quantificateurs absolus évaluent des mesures indépendamment du nombre d'individus dans l'univers considérés, et donc habituellement dans $\mathbb{R}^+$. Ces quantificateurs sont par exemple $MoinsDe3$, 5, $UneDizaine$ etc. (cf. Fig. 6).

Par exemple, si 5 jeunes sont grands parmi 6, les quantificateurs $LaPlupart$ et 5 conviennent. Si 5 jeunes sont grands parmi 100, le quantificateur absolu 5 s'applique toujours, mais le quantificateur relatif Peu est plus approprié. La notion de "quantificateur approprié" est détaillée ci-dessous dans le paragraphe sur le calcul de la valeur de vérité.

Trois formes de quantificateurs émergent parmi ceux présentés :

- trapèzes : $EnvironLaMoitié$ et $UneDizaine$ sont des trapèzes. 5, qui a la forme d'un triangle, est également considéré comme un trapèze particulier dont le noyau est réduit à un point.
- en Z : Peu et $MoinsDe3$
- en S : $LaPlupart$

Aucun autre type de quantificateur n'est considéré dans ce projet.

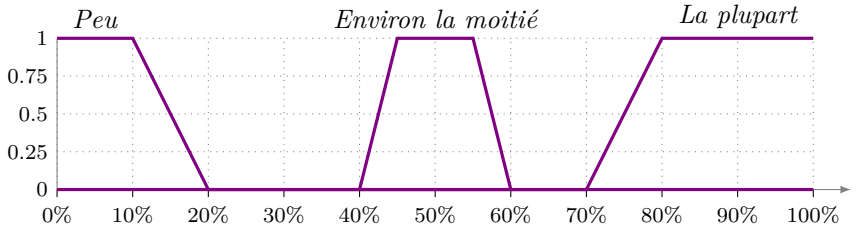

FIGURE 5 – Les trois quantificateurs relatifs "Peu", "Environ la moitié" et "La plupart"

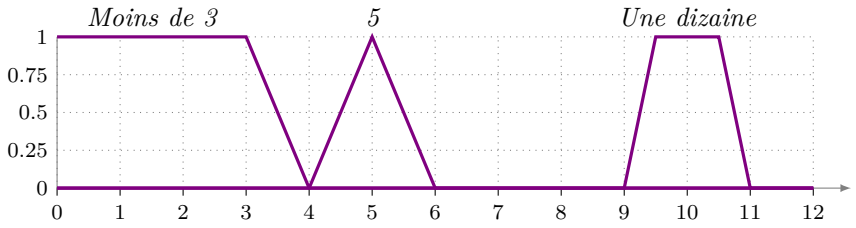

FIGURE 6 – Les trois quantificateurs absolus "Moins de 3", "5", et "Une dizaine"

Univers L'univers désigne l'ensemble des individus considérés pour le décompte. Dans l'exemple, l'univers considéré est celui des *Jeunes*, modalité d'une VL qui peut être définie de manière analogue à la VL *Taille* illustrée sur la Fig. 2.

Attribut Attribut mesuré sur les individus de l'univers. Dans l'exemple, c'est la VL *Taille* décrite sur la Fig. 2.

Partitions floues Une partition floue est un ensemble de sef tel que leur union recouvre l'ensemble partitionné. Trois types de partitions sont considérées dans le projet :

1. les partitions générales, qui n'ont aucune contrainte particulière, comme par exemple les modalités $MoinsDe3$ et 5 sur $[0;6]$
2. les partitions de Ruspini, telles que les m modalités $A_1 ... A_m$ d'une variable linguistique $A \in [a^-, a^+]$ vérifient $\forall x \in [a^-, a^+], \sum_{j=1...m} A_j(x) = 1$. Ces partitions ont certaines propriétés intéressantes en termes de calcul (cf. 6.1.1)
3. les partitions de Ruspini uniformes [6, Sec. 7], qui en plus de la contrainte ci-dessus sont égales par translation sur l'axe des x. Les modalités de la VL *Taille* représentées sur la Fig. 2 forment une partition de Ruspini uniforme coupée aux bornes de l'univers de définition.

2.2.2 Calcul de la valeur de vérité

Définition des fonctions d'appartenance La valeur de vérité mesure à l'adéquation d'un quantificateur Q à un décompte d'individus de l'univers R possédant l'attribut P. Pour mesurer cette adéquation, les fonctions d'appartenance décrivant les sef impliqués doivent être précisées.

Les sef utilisés dans le projet sont des trapèzes ou des triangles définis linéairement par parties. Par exemple, les quantificateurs $LaPlupart$ et $MoinsDe3$ sont notés Q_{LP} et $Q_{<3}$ et définis comme :

$$LaPlupart : [0,1] \to [0,1] \qquad MoinsDe3 : \mathbb{R}^+ \to [0,1]$$

$$x \to \begin{cases} 0 & \text{si } x \leq 0,7 \\ \dfrac{x-0,7}{0,1} & \text{si } x \in]0,7;0,8[\\ 1 & \text{si } x \geq 0,8 \end{cases} \qquad x \to \begin{cases} 1 & \text{si } x \leq 3 \\ 4-x & \text{si } x \in]3;4[\\ 0 & \text{si } x \geq 4 \end{cases}$$

Les modalités de $Taille$ sont notées $Petit(x)$, $Moyen(x)$, $Grand(x)$ et définies comme :

$$Petit : \mathbb{R} \to [0,1] \qquad Moyen : \mathbb{R} \to [0,1] \qquad Grand : \mathbb{R} \to [0,1]$$

$$x \to \begin{cases} 1 & \text{si } x \leq 90 \\ \dfrac{120-x}{30} & \text{si } x \in]90;120[\\ 0 & \text{si } x \geq 120 \end{cases} \quad x \to \begin{cases} 0 & \text{si } x \leq 90 \\ \dfrac{x-90}{30} & \text{si } x \in]90;120[\\ 1 & \text{si } x \in [120;150] \\ \dfrac{180-x}{30} & \text{si } x \in]150;180[\\ 0 & \text{si } x \geq 180 \end{cases} \quad x \to \begin{cases} 0 & \text{si } x \leq 150 \\ \dfrac{x-150}{30} & \text{si } x \in]150;180[\\ 1 & \text{si } x \geq 180 \end{cases}$$

Les modalités de Age, non illustrées ici mais définies de manière similaire à celles de $Taille$, sont notées

$Jeune(x)$, $Adulte(x)$, $Senior(x)$ et définies comme :

$$Jeune : \mathbb{R} \to [0,1] \qquad Adulte : \mathbb{R} \to [0,1] \qquad Senior : \mathbb{R} \to [0,1]$$

$$x \to \begin{cases} 1 & \text{si } x \leq 25 \\ \dfrac{30-x}{10} & \text{si } x \in]25;35[\\ 0 & \text{si } x \geq 35 \end{cases} \quad x \to \begin{cases} 0 & \text{si } x \leq 25 \\ \dfrac{x-30}{10} & \text{si } x \in]25;35[\\ 1 & \text{si } x \in [35;55] \\ \dfrac{60-x}{10} & \text{si } x \in]55;65[\\ 0 & \text{si } x \geq 65 \end{cases} \quad x \to \begin{cases} 0 & \text{si } x \leq 35 \\ \dfrac{x-60}{10} & \text{si } x \in]55;65[\\ 1 & \text{si } x \geq 65 \end{cases}$$

Calcul de la valeur de vérité Le protoforme "QRx sont P" s'instancie avec le quantificateur *LaPlupart*, dans l'univers *Jeunes* pour l'attribut *Grand* en la phrase "La plupart des jeunes sont grands". Sa valeur de vérité t est calculée comme suit :

$$t = Q\left(\frac{\sum_{x \in X} \top(R(x), P(x))}{\sum_{x \in X} R(x)}\right)$$

où X représente les données à résumer et $\top$ est une t-norme parmi $\top_Z$, $\top_L$ et $\top_P$.

Si le quantificateur Q est absolu, comme $MoinsDe3$ par exemple, alors la valeur de vérité s'exprime ainsi :

$$t = Q\left(\sum_{x \in X} \top(R(x), P(x))\right)$$

Les deux expressions se décomposent en 2 parties : le décompte à l'intérieur de la parenthèse, et la quantification, qui est l'application de Q au décompte.

Pour le quantificateur absolu, le décompte correspond au nombre de x qui sont à la fois R et P, i.e. ceux qui sont dans l'univers R et qui vérifient l'attribut P. Pour le quantificateur relatif, le décompte est ramené au nombre d'individus de l'univers, calculé ici comme la cardinalité floue de R.

Enfin, l'appartenance du décompte au quantificateur Q renvoie la valeur de vérité de la phrase.

Le paragraphe suivant donne un exemple de calcul de la valeur de vérité.

Exemple de calcul de la valeur de vérité Supposons que les données du Tableau 1 soient celles que nous souhaitons résumer, i.e. l'ensemble X.

X	Taille (cm)	Âge (années)
x_1	172	68
x_2	178	24
x_3	171	64
x_4	181	28
x_5	188	21

Tableau 1 – Exemple de données

Calculons la valeur de vérité de "Moins de 3 jeunes sont grands" par étapes :

1. D'abord, l'appartenance de chacun des x aux modalités considérées, à savoir *Jeune* et *Grand*, est calculée. Le résultat est donné dans le Tableau 2. Par exemple, $Grand(172) = (172 - 150)/30 \approx 0,73$.
2. Ensuite, la t-norme de *Jeune* et *Grand* est calculée, en utilisant la t-norme de Zadeh. Le résultat est également donné dans le Tableau 2. Par exemple, $\top_Z(Grand(172), Jeune(68)) = \min(0,73; 0) = 0$.
3. La somme sur l'ensemble des x est calculée. Le résultat 2,63, est indiqué dans la dernière ligne de la colonne $\top_Z(Jeune(x), Grand(x))$. Ainsi, au regard des définitions que nous avons prises pour la taille et l'âge, et en fonction des données disponibles, nous pouvons dire que 2,63 jeunes sont grands dans cette base. Le quantificateur sert justement à rendre ce résultat plus intelligible.
4. La dernière étape correspond au calcul de l'appartenance du décompte au quantificateur $MoinsDe3$. En l'occurrence, $MoinsDe3(2,63) = 1$, ce qui signifie que "Moins de 3 jeunes sont grands" est absolument vrai dans cette base et pour ces définitions.

Le calcul de "La plupart des jeunes sont grands" est similaire, avec simplement l'ajout du calcul de $\sum_x R(x)$, dont le résultat 2,7 est donné en bas de la colonne $Jeune(x)$. Le ratio $\sum_x \top_Z(Jeune(x), Grand(x))/\sum_x Jeune(x)$

donne alors $2,63/2,7 \approx 97\%$, ce qui signifie que 97% des jeunes de la base sont grands. L'appartenance de 97% au quantificateur *LaPlupart* est égale à 1, qui est donc la valeur de vérité de "La plupart des jeunes sont grands".

Si nous avions calculé la valeur de vérité de "Une dizaine de jeunes sont grands", le résultat aurait été 0 car $UneDizaine(2,67) = 0$ comme illustré sur la Fig. 6.

X	Taille (cm)	$Grand(x)$	Âge (années)	$Jeune(x)$	$\top_Z(Jeune(x), Grand(x))$
x_1	172	0,73	68	0	0
x_2	178	0,93	24	1	0,93
x_3	171	0,7	64	0	0
x_4	181	1	28	0,7	0,7
x_5	188	1	21	1	1
				$\sum = 2,7$	$\sum = 2,63$

Tableau 2 – Degrés d'appartenance des x à $Grand$ et $Jeune$

3 Principe de la solution

L'objectif de ce projet est la réalisation d'un système d'évaluation de résumés linguistiques flous. Le système prend en entrée une base de données et renvoie toutes les phrases possibles avec leur degré de vérité vérifiant certaines contraintes.

Le système doit être le plus simple possible d'utilisation. Les seuls éléments indispensables sont le fichier de données et celui des quantificateurs. Si l'utilisateur fourni un vocabulaire, il est utilisé, mais s'il n'en fourni pas, il est généré. Tout ou partie du vocabulaire peut être fourni, l'autre partie étant générée.

De plus, pour des raisons de simplicité, aucune IHM n'est demandée, tout est réalisé en ligne de commande et l'ensemble des fichiers en entrée et en sortie sont au format CSV.

De manière schématique, les grands composants du système sont :

- Préparation du vocabulaire : création des VL et des quantificateurs pour l'analyse de données
- Calcul des résumés : calcul des appartenances de chacune des colonne de données aux VL définies
- Génération des résumés : calcul des valeurs de vérité pour les différents protoformes considérés

Les sections suivantes décrivent les entrées / sorties du système puis ses grands composants, suivis d'une section dédiée aux différentes optimisations proposées.

4 Entrées / sorties du système

Ses entrées sont :

- Fichiers CSV
  - X : données
  - $Liens$: liens entre VL et colonnes dans les données
  - VL : variables linguistiques
  - $Quantif$: quantificateurs
- Paramètres (passés en ligne de commande - insensibles à la casse)
  - t_s : seuil de valeur de vérité
    - Les résumés dont les valeurs de vérité sont inférieures à t_s ne sont pas renvoyés. Le paramètre permet également d'accélérer leur génération.
    - Valeur par défaut : 0
  - *supp* : valeur de support minimal
    - aussi appelée Degree of Focus [7]
    - permet notamment d'optimiser les recherches dans l'arbre des résumés.
  - m_{def} : nombre de modalités par défaut
    - Utilisé pour générer des variables linguistiques à partir de leur min et de leur max

  - Valeur par défaut : 5
- fuz : degré de flou utilisé pour générer les modalités
  - Compris entre 0 et 1. Si $fuz = 0$, les modalités générées sont crisp, i.e. leurs noyaux et leurs supports sont identiques. Si $fuz = 1$, elles sont triangulaires, i.e. leurs noyaux sont des singletons. Si $0 < fuz < 1$, les modalités sont trapézoïdales, i.e. le nombre d'éléments de leurs noyaux est supérieur à 1 mais inférieur à celui de leurs supports.
  - Valeur par défaut : 0.5
- n : nombre de lignes de données
  - Si ce nombre n'est pas donné (valeur -1), le système le calcule.
  - Valeur par défaut : -1
- *skip* : nombre de lignes à ignorer en début de fichier de données
  - Permet par exemple d'ignorer une ou plusieurs lignes d'en-tête dans les données
  - Valeur par défaut : 0
- $tNorm$: t-norme à utiliser
  - peut être Z pour Zadeh, P pour probabiliste ou L pour Lukasiewicz
  - Valeur par défaut : P
- *entities* : nom des individus considérés (cf. section 7).
- $verbBe$: verbe être à utiliser (cf. section 7).
- $verbHave$: verbe avoir à utiliser (cf. section 7).
- $dataFormat$: format du fichier de données
  - FR si les points-virgules (;) servent à séparer les colonnes et les virgules (,) les parties entières des parties décimales
  - EN si les virgules (,) servent à séparer les colonnes et les points (.) les parties entières des parties décimales
  - valeur par défaut : EN
- $liensFormat$: comme $dataFormat$, pour le fichier structure
- $vocabFormat$: comme $dataFormat$, pour le fichier vocabulaire

Ses sorties sont :

– Fichier CSV
  - Sum : résumés avec leurs valeurs de vérité

5 Préparation du vocabulaire

Le vocabulaire recouvre :

– les VL utilisées pour décrire les données
– les liens avec les attributs (colonnes) des données
– les quantificateurs utilisés pour les résumés

Il peut être soit entièrement généré par le système, soit fourni, totalement ou partiellement, par l'utilisateur. Dans tous les cas, à l'issue de l'étape de préparation du vocabulaire, chaque attribut des données doit être associé à une VL. Si aucun quantificateur n'est précisé, ces derniers sont générés lors du rendu linguistique.

5.1 Généralités

– min : valeur minimale de l'univers de définition de la VL
– max : valeur maximale de l'univers de définition de la VL
– m : nombre de modalités de la VL
– d'une manière générale, pour les champs contenant des caractères |, si plusieurs | se suivent, un seul est retenu, et les | en début et en fin de champ sont ignorés
– de même, lorsque des valeurs numériques sont attendues, les caractères non numériques sont ignorés

Nom	Type	Modalités	Définition	Repr. R	Repr. P
vlTaille	C	Petit|Moyen|Grand	90|120|150|180	%entities% petites| %entities% de taille moyenne| %entities% grandes	%verbBe% petites| %verbBe% de taille moyenne| %verbBe% grandes
vlAge	C	Jeune|Adulte|Senior	25|35|55|65		
vlSalaire	C	Faible|Standard|Élevé	min/max 0|50000	%entities% ayant un salaire %modality%	%verbHave% un salaire %modality%
vlSexe	D	Homme|Femme	H((1|1))F((2|1))		
vlSaison	D	Hiver|Printemps|Été|Automne	Jan((1|1))Fev((1|1)) Mar((1|0,6)(2|0,4))...		
vlPoids	C		10|200		
vlTension	C				

Tableau 3 – Exemple de fichier de VL

5.2 Fichier de VL

Les VL fournies par l'utilisateur sont contenues dans un fichier CSV où chaque ligne représente une VL. Le Tableau 3 en donne un exemple.

5.2.1 Format du fichier de VL

Le fichier est réparti en 3 colonnes :

– "Nom" : contient le nom de la VL

- seule colonne obligatoire
- erreur possible : nom déjà existant

– "Type" : type de la VL, "C" pour continue et "D" pour discrète

- erreur possible : valeur autre que "C" ou "D"

– "Modalités" : contient les noms des différents modalités séparées par des pipes (|).

- si non renseigné, les noms de modalités sont construits sur le modèle $NomVL\_i$, où $NomVL$ est le nom issu de la 1ère colonne et $i = 1...m_{def}$. m est alors égal à m_{def}

– "Définition" : dans le cas d'une VL continue, contient les points clés d'une VL Ruspini ou les valeurs de min et max pour une VL Ruspini uniforme, dans le cas d'une VL discrète, correspond les valeurs associées à chaque modalité.

- VL continue :
  - si la définition commence par "min/max", les deux valeurs sont le min et le max de la VL
    - erreur possible : nombre de valeurs différent de 2 ou $max \leq min$
  - sinon, ce sont les points clés de la VL
    - par exemple, pour la VL $Taille$ illustrée Fig. 2, les points clés sont 90, 120, 150 et 180 : 90 marque la fin du noyau de la 1ère modalité et le début du support de la 2ème, 120 la fin du support de la 1ère et le début du noyau de la 2ème, 150 la fin du noyau de la 2ème et le début du support de la 3ème etc.
    - erreur possible : nombre de valeurs différent de $2(m-1)$
  - si "Modalités" est renseigné, alors les valeurs sont les points clés correspondant
  - sinon, les valeurs sont le min et le max de la VL
- VL discrète
  - Format : valeur de l'univers((N°modalité\|Appartenance)(N°modalité\|Appartenance)...)...

— Par exemple, la chaîne complète de définition de la VL discrète *Saison* décrite sur la Fig. 3 est Jan((1|1))Fev((1|1))Mar((1|0,6)(2|0,4))Avr((2|1))Mai((2|1))Jun((2|0,6)(3|0,4))Jui((3|1)) Aou((3|1))Sep((3|0,6)(4|0,4))Oct((4|1))Nov((4|1))Dec((4|0,6)(1|0,4)), sachant que les modalités sont définies dans l'ordre Hiver=1, Printemps=2 etc...

— Noter que les modalités qui ne sont pas définies pour une valeur donnée de l'univers valent 0 pour cette valeur

— erreur possible :

— N° modalité < 1 ou $> m$

— Appartenance < 0 ou > 1

— Somme des appartenances différente de 1 pour une valeur de l'univers

- Les représentations R et P sont détaillées dans la section 7. Si une seule représentation est donnée (comme sur la $vlSalaire$), elle s'applique pour toutes les modalités. Si plusieurs formes sont données (comme sur la $vlTaille$), elles s'appliquent à chacune des modalités

— erreur possible : nombre de formes différents du nombre de modalités

5.2.2 Exemple de résultat attendu

A la lecture du Tableau 3, le système doit :

– créer deux VL continues $vlTaille$ et $VlAge$ à partir des définitions complètes données dans les 2 premières lignes

– créer une VL continue $vlSalaire$ générée comme une partition uniforme de Ruspini entre 0 et 50000 recherchés dans les données et 3 modalités $Faible$, $Standard$, $Elevé$

– créer une VL discrète $vlSexe$, composée de 2 modalités $Homme$ et $Femme$ telles que $Homme(H) = 1$, $Homme(F) = 0$, $Femme(H) = 0$ et $Femme(F) = 1$

– créer une VL discrète $vlSaison$, composée de 4 modalités $Hiver$, $Printemps$, $Eté$ et $Automne$ et telle que décrite dans la Fig. 3

– créer une VL continue $vlPoids$ générée comme une partition uniforme de Ruspini entre 10 et 200 avec m_{def} modalités nommées $vlPoids\_1$ à $vlPoids\_m_{def}$

– créer une VL continue $vlTension$ générée comme une partition uniforme de Ruspini entre min et max recherchés dans les données et m_{def} modalités nommées $vlTension\_1$ à $vlTension\_m_{def}$

5.3 Fichier de quantificateur

Nom	Type	Forme	Définition	Représentation
LaPlupart	R	S	0.7|0.8	La plupart des
Peu	R	Z	0.1|0.2	Peu de
Environ50%	R	T	0.4|0.45|0.55|0.6	Environ 50% des
MoinsDe3	A	Z	3|4	Moins de 3
5	A	T	4|5|6	5
UneDizaine	A	T	9|9,5|10,5|11	Une dizaine de

Tableau 4 – Exemple de fichier de Quantificateurs

5.3.1 Format du fichier de Quantificateur

Le fichier est réparti en 5 colonnes :

– "Nom" : contient le nom du quantificateur

- si non renseigné, les noms sont générés sur le format "Q_1", "Q_2" etc.

– "Type" : type du quantificateur

- "A" pour absolu, "R" pour relatif.
- colonne obligatoire
- erreur possible : valeur différente de "A" ou "R"

– "Forme" : forme du quantificateur
  - "S" pour les quantificateurs en S, "Z" pour ceux en Z, et "T" pour ceux en trapèze (cf. 2.2.1)
  - colonne obligatoire
  - erreur possible : valeur différente de "S", "Z" ou "T"
– "Définition" : donne les paramètres du quantificateur en fonction de sa forme
  - colonne obligatoire
  - si "S" ou "Z", deux valeurs attendues
  - si "T", quatre valeurs attendues
  - erreur possible : nombre de valeurs attendues non trouvé
– "Représentation" : représentation linguistique (cf. section 7)
  - si la représentation n'est pas donnée, le nom est utilisé

5.3.2 Exemple de résultat attendu

A la lecture du Tableau 4, le système doit :
– créer un quantificateur relatif en S nommé "LaPlupart"
– etc.

5.4 Fichier de liens

Le fichier de liens renseigne les liens entre les VL et les colonnes en base. Il est composé de lignes avec le numéro de la colonne dans les données et la VL correspondante séparée par des virgules. Le tableau 5 donne un exemple d'un tel fichier.

Colonne	VL
1	vlTaille
2	vlAge
3	vlSexe

Tableau 5 – Exemple de fichier de Liens

Erreurs possibles :
– numéro de colonne < 0
– numéro de colonne $>$ nombre de colonnes déterminé dans le fichier de données

6 Calcul des résumés

L'objectif du calcul des résumés est de récupérer l'ensemble des valeurs de vérité pour les VL/modalités et les combinaisons de VL/modalités qui permettront de générer les phrases par la suite.

Le calcul des résumés est effectué en 4 étapes :

1. 1ère passe sur les données permettant de calculer les cardinalités de chaque modalité de chaque VL
2. 2ème passe sur les données permettant de calculer les cardinalités de chaque combinaisons de modalités de VL
3. La valeur de vérité des protoformes "Qx sont P"
4. La valeur de vérité des protoformes "QRx sont P"

6.1 1ère passe

Lors de la 1ère passe, détaillée dans l'algorithme 1, le fichier de données est lu depuis le début, et pour chaque ligne, l'appartenance de chaque valeur est testée avec la VL qui lui est associée. La ou les appartenances à la ou les modalités sont ajoutées aux appartenances déjà calculées dans une structure du type tableau à 2 entrées, notée $card(j, k)$, où j représente le n° de VL et k le n° de modalité dans cette VL.

Les modalités de calcul rapide de l'appartenance d'une valeur à 1 ou 2 modalités d'une partition sont données dans les deux paragraphes suivants.

Algorithme 1 Lecture des données

```
Pour tout j,k, card(j,k)=0
Pour chaque ligne l du fichier de données  # Parcours du fichier
  Pour chaque colonne j de l               # Parcours des attributs
    mShips(j) = VL(j).mShip(l(j))         # Calcul de la valeur d'appartenance
    card(j,k1) += mu1                     # Prise en compte la 1ère modalité
    si nMod > 1 card(j,k2) += mu2         # Si 2, prise en compte la 2ème
```

6.1.1 Calcul rapide de l'appartenance de x à une partition de Ruspini

Données décrivant la partition m le nombre de modalité, min la plus petite valeur de la VL, max la plus grande valeur de la VL, P l'ensemble des points clés notés p_l pour $l = 1...p$, avec $p = 2(m-1)$.

Appartenance de x à l'une des modalités Plutôt que de parcourir chaque modalité séquentiellement, une stratégie diviser / conquérir est envisagée, comme indiqué sur l'algorithme 2.

Algorithme 2 Appartenance dans une partition de Ruspini

```
if x < p_1 return (n_k = 1, k_1 = 1, μ_1 = 1)                    # Cas triviaux
if x > p_P return (n_k = 1, k_1 = P, μ_1 = 1)
leftId ← 1, rightId ← p                                          # Initialisation
i ← ⌊(rightId − leftId)/2⌋
while x ∉ [p_i, p_{i+1}]                                         # Boucle
  if x < p_i  rightId ← i
  else leftId ← i
  i ← ⌊(rightId − leftId)/2⌋
if i mod 2 = 0 return (n_k = 1, k_1 = i/2, μ_1 = 1)              # Dans un noyau
else return (n_k = 2, k_1 = (i+1)/2, μ_1 = (p_{i+1} − x)/(p_{i+1} − p_i),  # Entre 2 modalités
    k_2 = k_1 + 1, μ_2 = 1 − μ_1
```

Complexité En $log_2(m)$ puisque les modalités sont ordonnées et que la stratégie diviser/conquérir est utilisée.

6.1.2 Calcul rapide de l'appartenance de x à une partition de Ruspini uniforme

Données décrivant la partition m le nombre de modalité, min la plus petite valeur de la VL, max la plus grande valeur de la VL et fuz le degré de flou retenu.

Variables calculées ν la taille d'un noyau, δ la distance entre 2 noyaux successifs, définies tels que :

$$\nu = (1 - fuz)\frac{max - min}{m - 2} \qquad \delta = fuz\frac{max - min}{m - 1}$$

Appartenance de x à l'une des modalités Comme la partition considérée est de Ruspini, x appartient à 1 ou 2 modalités de la VL A, dont les indices sont notés k_1 et k_2, avec $1 \leq k_1, k_2 \leq m$. Soit $n_k \in \{1, 2\}$ le nombre de modalités auxquelles x appartient. Si $n_k = 1$, *seul* k_1 est défini. Si $x < min$, alors $n_k = 1$ et $k = 1$. Si $x > max$, alors $n_k = 1$ et $k = m$.

Sinon, nous introduisons la variable auxiliaire ϕ définie ainsi :

$$\phi = (x - min) \mod (\nu + \delta)$$

permettant le calcul des autres valeurs :

$$n_k = \begin{cases} 1 & \text{si } \phi \geq \delta \\ 2 & \text{sinon} \end{cases} \qquad k_1 = \left\lceil \frac{x - min}{\nu + \delta} \right\rceil + 1\,(\phi \geq \delta) \qquad k_2 = k_1 + 1$$

$$A_{k_1}(x) = \begin{cases} 1 & \text{si } n_k = 1 \\ 1 - \dfrac{\phi}{\delta} & \text{sinon} \end{cases} \qquad A_{k_2}(x) = 1 - A_{k_1}(x)$$

avec $1(\phi \geq \delta) = 1$ si $\phi \geq delta$ et 0 sinon.

Complexité $O(1)$, puisque le calcul de la ou des valeurs d'appartenance ne dépend pas du nombre de modalités à tester. Il n'est pas possible de calculer plus rapidement l'appartenance d'une valeur à une partition floue, c'est pourquoi les partitions de Ruspini uniformes sont à privilégier dans ce contexte.

6.2 2ème passe

La 2ème passe permet le calcul des de l'appartenance des données à des combinaisons de modalités, par exemple *Jeune* et *Grand*. Formellement, cela revient à remplir la structure $card(j,k,j',k') = \top(A_{jk}, A_{j'k'})$ oùA_{jk} représente la $k^{ème}$ modalité de la VL A_j. Comme la t-norme est symétrique, les seules valeurs de $card(j,k,j',k')$ à calculer sont telles que $j' > j$.

La 2ème passe est décrite par l'algorithme 3.

Algorithme 3 2ème passe

```
Pour tout j,k,j',k' card(j,k,j',k')=0
Pour chaque ligne l du fichier de données          # Parcours du fichier
  Pour chaque colonne j de l                       # Calcul des combinaisons
    Pour chaque j' > j
      si card(j,k1) > supp et card(j',k1') > supp
        card(j,k1,j',k1') += tNorm(mu1, mu1')
      si card(j,k2) > supp et card(j',k1') > supp et nMod=2
        card(j,k2,j',k1') += tNorm(mu2, mu1')
      si card(j,k1) > supp et card(j',k2') > supp et nMod'=2
        card(j,k1,j',k2') += tNorm(mu1, mu2')
      si card(j,k2) > supp et card(j',k2') > supp et nMod=2 et nMod'=2
        card(j,k2,j',k2') += tNorm(mu21, mu2')

Note: la structure mShips(j) a déjà été calculée dans l'algorithme 1.
nMod=mShips(j).nMod, nMod'=mShips(j').nMod
k1=mShips(j).k1, k1'=mShips(j').k1, k2=mShips(j).k2, k2'=mShips(j').k2
mu1=mShips(j).mu1, mu1'=mShips(j').mu1, mu2=mShips(j).mu2, mu2'=mShips(j').mu2
```

6.3 Calcul des résumés "Qx sont P"

A l'issue des 2 passes, les valeurs de vérité de différents résumés peuvent être calculées. Dans un premier temps, c'est celle des résumés "Qx sont P" qui est évaluée, comme indiqué sur l'algorithme 4).

Pour cela, il convient de calculer l'appartenance de chacune des cardinalités avec les quantificateurs fournis entrée, éventuellement divisée par le nombre de données dans le cas des quantificateurs relatifs. Le résumé est ajouté à la liste des résumés à renvoyer uniquement si sa valeur de vérité est supérieure à t_s.

Algorithme 4 Calcul des résumés "Qx sont P"

```
Pour chaque quantificateur q donné en entrée
  Pour chaque VL j
    Pour chaque modalité k de j telle que card(j,k)>supp  # Support suffisant ?
      si q.type = Absolute t = q.mShip(card(j,k))         # Val. vérité - cas Q Absolu
      sinon t = q.mShip(card(j,k) / n)                    # Val. vérité - cas Q Relatif
      si t > ts resumesQP.ajoute(q,j,k,t)                 # > seuil ? ajoute à la liste
```

6.4 Calcul des résumés "QRx sont P"

La valeur de $card(j,k,j',k')$, combinée avec celles de $card(j,k)$ et $card(j',k')$, permet de calculer les valeurs de vérité de :

- $t(QA_{jk} \text{ sont } A_{j'k'}) = \dfrac{card(j,k,j',k')}{card(j,k)}$

– $t(QA_{j'k'} \text{ sont } A_{jk}) = \dfrac{card(j,k,j',k')}{card(j',k')}$

Deux modalités de la même VL ne sont pas comparées : cela reviendrait à tester "La plupart des grands sont petits" par exemple, ce qui n'a pas d'intérêt. Cela permet également de ne pas avoir à vérifier la propriété $\top(A_j, A_k) = 0$ puisque ce cas n'est jamais évalué. Ainsi, $card(j,k,j',k')$ n'est évalué que pour $j \neq j'$.

De même une modalité n'est jamais comparée à elle-même, puisque la valeur de vérité d'une phrase comme "Tous les petits sont petits" n'a pas d'intérêt non plus. Cela permet également de ne pas avoir à vérifier la propriété $\top(A, A) = A$ puisque ce cas n'est jamais évalué.

Toutes les combinaisons n'ont pas à être testées du fait de la valeur *supp* de support minimal fournie en entrée : si $card(j,k)/n < supp$, le résumé n'a pas à être renvoyé. D'autres optimisations sont envisageables mais non détaillées ici. Elles sont toutefois évoquées dans la section 9.

Le génération des résumés est décrite par l'algorithme 5.

Algorithme 5 Calcul des résumés "QRx sont P"

```
Pour chaque VL j
  Pour chaque VL j' > j
    Pour chaque modalité k de la VL j
      Pour chaque modalité k' de la VL j'
        Pour chaque quantificateur q
          si q est absolu
            t = q.mShip(card(j,k,j',k'))
            si t > ts resumesQRP.ajoute(q,j,k,j',k',t)    #> seuil ? ajoute
          si q est relatif
            si card(j,k) > supp                           #support de QRP ok ?
              t = q.mShip(card(j,k,j',k')/card(j,k))
              si t > ts resumesQRP.ajoute(q,j,k,j',k',t)  #> seuil ? ajoute
            si card(j',k') > supp                         #support de QPR ok ?
              t = q.mShip(card(j,k,j',k')/card(j',k'))
              si t > ts resumesQRP.ajoute(q,j',k',j,k,t)  #> seuil ? ajoute
```

7 Génération des phrases

La génération des phrases permet la conversion des résumés calculés à l'étape précédentes en phrases du langage naturel.

Les deux langues prises en comptes dans ce projet sont le français et l'anglais, dont les phrases ont une structures identiques dans les contexte des résumés linguistiques flous.

7.1 Principe

Les différents cas à traiter sont :

1. Les modalités de R et P peuvent être utilisées directement pour générer la phrase
   – Ex : "la plupart des jeunes sont grands"
     - "la plupart" : nom de Q
     - "des" : article de Q
     - "jeunes" : modalité de R au pluriel
     - "sont" : verbe
     - "grands" : modalité de P au pluriel
2. La modalité de R ne peut être utilisée directement, mais la modalité de P signifie
   – Ex : "la plupart des individus qui ont un salaire élevé sont jeunes"
     - "la plupart" et "des" : nom et article de Q
     - "individus qui ont un salaire élevé" : nom des entités + "qui ont un " + nom VL + nom modalité
     - "sont" : verbe

- "jeunes" : modalité de P

3. La modalité de R peut être utilisée directement mais pas celle de P
   - Ex : "la plupart des jeunes sont de taille moyenne" vs. "la plupart des jeunes sont moyens"
     - "la plupart" et "des" : nom et article de Q
     - "jeunes" : modalité de R
     - "sont" : verbe
     - "de taille moyenne" : valeur particulière
4. Le verbe utilisé est différent
   - "La plupart des naissances en hiver sont des garçons " vs. "La plupart des garçons naissent en hiver"

La phrase est donc constituée de plusieurs blocs :

- Bloc général, i.e. associé au résumé
  - Verbes utilisés, représentés par `%verbBe%` et `%verbHave%`
  - Nom des entités étudiées, représenté par `%entities%`
- Bloc associé au quantificateur
  - Ex : "La plupart des" associé à *LaPlupart*, "Environ 5" à *Environ*5 etc...
- Blocs associés aux VL
  - Utilisées en qualifier
    - Ex : "`%entities%` de taille moyenne" pour la modalité *Moyen* de la VL *Taille*, dans une base de personnes avec `entities`=individus
    - Ex : "gros `%entities%`" pour la modalité *Gros* de la VL *Taille*, dans une base de fruits avec `entities`=fruits
    - Ex : "`%entities%` qui ont un salaire élevé" pour la modalité *Eleve* de la VL *Salaire*, dans une base de personnes avec `entities`=individus
  - Utilisées en attribut
    - Ex : "de taille moyenne" pour la modalité *Moyen* de la VL *Taille*
    - Ex : "gros" pour la modalité *Gros* de la VL *Taille*
    - Ex : "ont un salaire élevé" pour la modalité *Elevé* de la VL *Salaire*

La phrase finale est la concaténation du bloc quantificateur, du bloc qualifie, du bloc attribut et de la valeur de vérité entre parenthèses.

7.2 Exemple

La base de données une base d'individus, la langue du résumé est le français. Les données associées au résumé sont donc :

- `entities`=personnes
- `verb`Be=sont
- `verb`Have=ont

Les quantificateurs utilisés sont *Peu* et *LaPlupart*, associés aux représentations linguistiques "Peu de" et "La plupart des".

Les VL sont :

- *Taille* associée aux modalités *Petit*, *Moyen* et *Grands* elles mêmes rattachées aux représentations linguistiques suivantes :
  - En qualifier R :
    - *Petit* : "`%entities%` petites"
    - *Moyen* : "`%entities%` de taille moyenne"
    - *Grand* : "`%entities%` grandes"
  - En attribut P :

— *Petit* : "`%verbBe%` petites"
— *Moyen* : "`%verbBe%` de taille moyenne"
— *Grand* : "`%verbBe%` grandes "

– *Salaire* associée aux modalités *Faible*, *Correct* et *Elevé* elles mêmes rattachées aux représentations linguistiques suivantes :

- En qualifier R :

— Toutes modalités : "`%entities%` ayant un salaire `%modality%` "

- En attribut P :

— Toutes modalités : "`%verbHave%` un salaire `%modality%`"

Supposons qu'à l'étape le résumé constitué des éléments $Q = LaPlupart$, $VL\,R = Taille$, $Modalité\,R = Petit$, $VL\,P = Salaire$, $Modalité\,P = Correct$, a été renvoyé avec une valeur de vérité 0,83.

La phrase résultat est composée selon les étapes suivantes :

1. Bloc quantificateur : "La plupart des" est retrouvé directement depuis sa représentation linguistique
2. Bloc qualifier R : "`%entities%` petites" et `entities`=personnes, donc "personnes petites"
3. Bloc attribut P : "`%verbHave%` un salaire `%modality%`", `verbHave`=ont et `modality`=correct, donc "ont un salaire correct"
4. Concaténation des blocs et de la valeur de vérité : "La plupart des personnes petites ont un salaire correct (0.83)"

Supposons un autre le résumé constitué des éléments $Q = Peu$, $VL\,R = Salaire$, $Modalité\,R = Elevé$, $VL\,P = Taille$, $Modalité\,P = Petit$ avec une valeur de vérité 0,41.

La phrase résultat est composée selon les étapes suivantes :

1. Bloc quantificateur : "Peu de" est retrouvé directement depuis sa représentation linguistique
2. Bloc qualifier R : "`%entities%` ayant un salaire `%modality%` ", `entities`=personnes et `modality`=élevé donc "personnes ayant un salaire élevé"
3. Bloc attribut P : "`%verbBe%` petites" et `verb`Be=sont, donc "sont petites"
4. Concaténation des blocs et de la valeur de vérité : "Peu de personnes ayant un salaire élevé sont petites (0.41)"

8 Spécification technique

8.1 Diagramme de classes

Le digramme de classes du projet est présenté sur la Fig. 7.

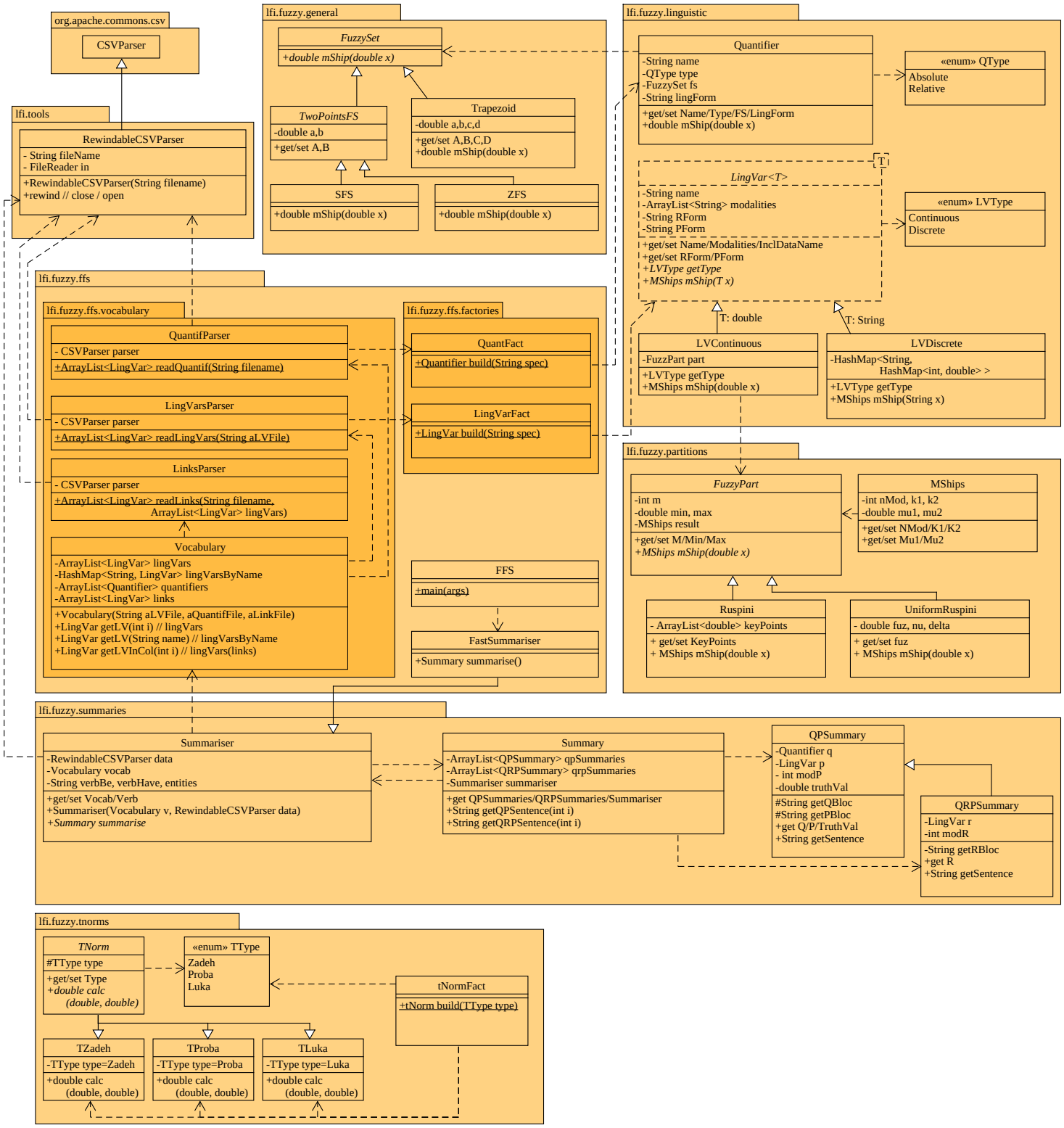

FIGURE 7 – Diagramme de classes

8.2 Package org.apache.commons.csv

Les lectures de fichiers CSV seront réalisées à l'aide de la bibliothèque Apache Commons CSV [2].

D'une manière générale, les types de données lus dans les fichiers sont vérifiés. Par exemple, dans le fichier de Quantificateurs, si le type n'est ni "A" ni "R", une exception est renvoyée. Ces exceptions sont ensuite affichées à l'utilisateur, afin de l'informer d'une erreur dans la définition de son fichier.

2. `http://commons.apache.org/proper/commons-csv/`

8.3 Package lfi.fuzzy.ffs

8.3.1 Classe Vocab

Lors de la construction des object LingVar, si une liste de keyPoints est donnée telle que les modalités ont toute la même forme, alors il faut créer un UniRuspini et non un Ruspini, car le premier est optimisé pour le calcul des valeurs d'appartenance.

9 Évolutions

- Arbitrer sur les quantificateurs : si aucun n'est fourni, en générer automatiquement - les définir précisément, voir l'impact sur les optimisations...
- Inclure la génération des résumés dans les passes 1 et 2 au lieu de calculer les cardinalités d'abord puis leur appartenance aux quantificateurs
- Étudier sous quelles conditions l'évaluation d'une VL peut être interrompue durant une passe, i.e. avant la lecture de toutes les données

  - si son support / sa valeur de vérité ne pourra atteindre un seuil

- Étendre les protoformes à QRx sont P_1 et P_2 et ... et P_n
- Autres évolutions : cf. optim.pdf

10 Complexité

10.1 Notations

- n : nombre de données (lignes)
- v : nombre de VL
- m : nombre de modalités par VL - par simplicité, on considère que toutes les VL ont le même nombre de modalités
- m_s : nombre de modalités par VL n'atteignant pas le support minimal - par simplicité, on considère que ce nombre est le même pour toutes les VL

10.2 Calcul pour des partitions générales

Dans ce mode de calcul, on suppose que l'appartenance de chaque donnée à chaque modalité de chaque VL doit être réalisé. En ce cas, le calcul de $card(j,k)$ est réalisé en mnv opérations.

Par la suite, on considère que les partitions des VL sont des Ruspini uniformes et que l'appartenance d'un x est calculé en 1.

10.3 Discussion sur le parcours des données

Est-il plus efficace de parcourir les données en 2 passes ou en 1 passe ?
Deux méthodes sont envisageables :

1. Parcours en 1 passe sur les données, où les structures $card(j,k)$ et $card(j,k,j',k')$ sont calculées
2. Parcours en 2 passes données, où la 1ère passe calcule $card(j,k)$ et la 2ème $card(j,k,j',k')$ en excluant les calculs pour les modalités dont la valeur n'atteint pas le support minimal

Les calculs des résumés "Qx sont P" et "QRx sont P" sont identiques quelle que soit l'option retenue (cf. plus loin).

Dans les deux cas, le calcul de $card(j,k)$ est de complexité identique, mais celui de $card(j,k,j',k')$ est modifié en fonction de la connaissance de la cardinalité des VL et donc de leur comparaison par rapport au seuil de support.

La Fig. 8 fournit une représentation des éléments devant être calculés pour la construction de $card(j,k,j',k')$. Le tableau représente les croisements entre VL. Les cellules blanches requièrent un calcul, les colorées non. Celles en rose n'ont pas à être calculées du fait de la symétrie de la t-norme utilisées pour les croisement, celles en vertes du fait de la non comparaison de modalités de VL entre elles, et celles en bleu indiquent des modalités dont la cardinalité n'atteint pas le support minimal. Les cellules blanches y compris les bleues sont au nombre de $m^2v(v-1)/2$. Les cellules bleues sont au nombre de m_s^2.

Le nombre de calcul à effectuer en tenant en compte les supports est donc $m^2v(v-1)/2 - m_s^2$.

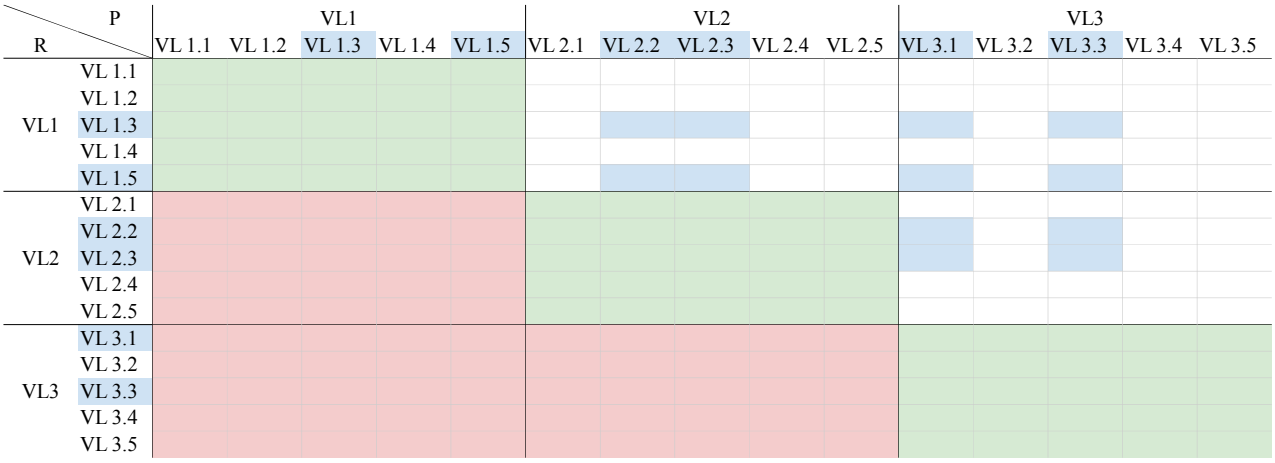

FIGURE 8 – Complexité du calcul de $card(j, k, j', k')$

Le parcours en 1 passe requiert donc, pour chaque ligne, le calcul de $card(j, k)$ sur les v attributs, soit v opérations, puis le calcul de tous les croisements possibles, donc $m^2v(v-1)/2$. La complexité du parcours en 1 passe est donc $nv(1 + m^2(v-1)/2)$.

Le parcours en 2 passe requiert 1 première passe en nv puis une seconde passe en $n(m^2v(v-1)/2 - m_s^2)$, soit une complexité totale de $nv(1 + m^2(v-1)/2 - m_s^2)$.

Il est donc clair que le parcours en 2 passes est plus rapide que le parcours en 1 passe, puisque $\forall m_s > 0$, $nv(1 + m^2(v-1)/2 - m_s^2) < nv(1 + m^2(v-1)/2)$.

Cette remarque ne se base bien sûr que sur la complexité, et pas sur les considérations hardware de lecture / écriture sur le disque, qui fait qu'un algorithme en 1 passe peut être plus rapide si le fichier de données est chargé depuis le disque et pas exclusivement lu depuis la mémoire.

10.4 Complexité du calcul des résumés

10.4.1 "Qx sont P"

TODO

10.4.2 "QRx sont P"

TODO

Références

[1] R. YAGER, « A new approach to the summarization of data », *Inf. Sci. (Ny).*, vol. 28, no. 1, p. 69–86, 1982.

[2] J. KACPRZYK et S. ZADROZNY, « Computing With Words is an Implementable Paradigm : Fuzzy Queries, Linguistic Data Summaries, and Natural-Language Generation », *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, p. 461–472, 2010.

[3] B. BOUCHON-MEUNIER, *La Logique Floue*. PUF, 2007.

[4] L. ZADEH, « Fuzzy sets », *Inf. Control*, vol. 8, no. 3, p. 338–353, 1965.

[5] L. ZADEH, « The concept of a linguistic variable and its application to approximate reasoning - I », *Inf. Sci. (Ny).*, vol. 8, no. 3, p. 199–249, 1975.

[6] R. MESIAR et A. STUPŇANOVÁA, « Open problems from the 12th International Conference on Fuzzy Set Theory and Its Applications », *Fuzzy Sets Syst.*, 2014.

[7] J. KACPRZYK et A. WILBIK, « Towards an efficient generation of linguistic summaries of time series using a degree of focus », *in Proc. of NAFIPS'09*, p. 1–6, 2009.